IOWA STATE UNIVERSITY
Digital Repository

Retrospective Theses and Dissertations

Iowa State University Capstones, Theses and Dissertations

1997

Applying virtual reality techniques to engineering design optimization

Tsung-Pin Yeh
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/rtd

 Part of the Computer Sciences Commons, and the Mechanical Engineering Commons

Recommended Citation
Yeh, Tsung-Pin, "Applying virtual reality techniques to engineering design optimization " (1997). *Retrospective Theses and Dissertations*. 11576.
https://lib.dr.iastate.edu/rtd/11576

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

Applying virtual reality techniques to engineering design optimization

by

Tsung-Pin Yeh

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Mechanical Engineering

Major Professor: Judy M. Vance

Iowa State University

Ames, Iowa

1997

Copyright © Tsung-Pin Yeh, 1997. All rights reserved.

UMI Number: 9814713

Copyright 1997 by
Yeh, Tsung-Pin

All rights reserved.

UMI Microform 9814713
Copyright 1998, by UMI Company. All rights reserved.

This microform edition is protected against unauthorized
copying under Title 17, United States Code.

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

Graduate College
Iowa State University

This is to certify that the Doctoral dissertation of

Tsung-Pin Yeh

has met the dissertation requirements of Iowa State University

Signature was redacted for privacy.

Major Professor

Signature was redacted for privacy.

For the Major Program

Signature was redacted for privacy.

For the Graduate College

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my major professor Dr. Judy M. Vance for her valuable guidance, support, and encouragement on this work and throughout my graduate studies. Her patience, understanding, and friendship are also appreciated.

I would also like to extend my appreciation to Dr. James E. Bernard, Dr. James H. Oliver, Dr. Donald R. Flugrad, Dr. Jerald M. Vogel, and Dr. Jo Min for serving on my doctoral committee as well as offering their knowledge and suggestions in the areas of sensitivity analysis, computer graphics, machine design, finite element analysis, and design optimization. Thanks are also due to Dr. Carolina Cruz-Neira for her inspiration and instructions in the field of virtual reality.

Furthermore, I would like to thank the support of my colleagues at the Iowa Center for Emerging Manufacturing Technology to help me along the way of my research.

Finally, I would like to give my deepest appreciation to my mother, Chiu-Yen Chong, and my beautiful wife, Hsui-Chen Yeh, for their unconditional support, sacrifice, and love.

ABSTRACT

Virtual reality (VR) provides a new paradigm that links a human operator with a computer-generated environment via visual, audio, or haptic interaction to support the illusion of immersion. The objective of this dissertation is to explore the use of VR techniques for sensitivity-based structural shape design optimization. The design tool developed in this research allows the designer to alter the shape of the computer representation of a model and see the resulting changes to the deformation and stress displayed immediately in the virtual environment. The combination of using natural motions to alter computer models and the ability to view stereo images of these models in a three dimensional virtual space allows the designer to interactively examine design changes and determine their effects on product performance.

NURBS-based free form deformation (NFFD) and direct manipulation techniques are the methods implemented which allow the designer to change the shape of the design model, and also provide the information for the sensitivity computation in order to approximate the structural responses in real time. The designer-weighted optimization is also incorporated in the virtual environment which provides an additional capability to perform trade-off design for resolving conflicting design constraints.

A general purpose structural shape design program is developed to perform the structural shape design optimization in the virtual environment. Several examples demonstrate the advantages of this program.

CHAPTER 1. INTRODUCTION

Virtual reality (VR) provides a design space consisting of three-dimensional computer images where participants can interact with these images using natural human motions in real time. In the field of engineering design, prototyping and design verification have provided the initial application area for VR. The research presented in this dissertation takes the scenario one step further by incorporating geometric free-form deformation, finite element analysis, sensitivity methods, and interactive design optimization techniques into the virtual world such that the designer can easily implement analysis-based shape design optimization of a structural system where deformation and stress consideration are of interest. Using immersive VR techniques as the interactive design tool provides a natural and easy way to perform shape design with analysis considerations, which will give the designer an intuitive sense of the performance of the design and allow the designer to achieve a viable, optimal solution in a timely manner.

1.1 Motivation

The research is driven by the desire to investigate the potential of virtual reality techniques to provide the logical evolution of existing human-computer interfaces. VR removes the barriers of the traditional computer interface which consist of the keyboard, mouse and monitor, and allows the user to experience a strong sense of presence in a three-dimensional computer-generated scene. It enhances the user's role in the human-machine interaction scenario in the following ways:

1. VR provides an easy and intuitive method to navigate among the computer models by using head movement to control the viewpoint, just as in real life when the user views a physical model. This capability provides the user with better spatial understanding of the three-dimensional geometry of the computer models that is not provided by traditional computer interfaces.
2. Natural hand motion can be used to manipulate virtual objects in VR. By using an input device, such as an instrumented glove, a three-dimensional mouse or even audio input, the designer is able to design and manipulate computer models as if they exist as real objects.
3. VR offers a stereo view of the computer model which further extends the user's understanding of the three-dimensional model. The stereo view is created by using offset images to simulate the binocular depth cues of convergence and stereopsis of the user's eyes. The depth cues provided by the stereo image are very helpful in conveying the spatial relationships of the computer-generated images.
4. Other VR peripheral such as haptic feedback, sound feedback, and vocal command input can enhance the ability of the user to interact with the virtual objects.

Combined with the rapid advancement of computer hardware and graphics software technologies, VR has advanced from a research curiosity toward application as a serious new tool in a variety of fields. VR has shown promising contributions in architectural walk-throughs, medical research, training and education, telepresence, and entertainment [1]. Currently, the most common use of VR in engineering is for prototyping, design visualization, assembly verification, and conceptual design [2][3]. This research focuses on the use of VR in analysis-based design for structural systems. By incorporating the geometric manipulation

techniques, analysis results and sensitivity information into the virtual environment, VR holds the potential to become an innovative design and analysis tool such that the designer can perform the geometric shape changes with analysis consideration all inside the virtual environment.

1.2 Objective

The objective of this research is to develop a virtual environment to facilitate the interactive design optimization of structural systems. The designer will be able to take advantage of VR techniques to change the shape of a virtual object using natural hand motion and gestures. With a stereo view of the object, the designer can view the 3D computer image just like a physical model placed in front of the designer. In order to perform structural shape design of a virtual object, a mechanism to relate the structural shape design variables and the finite element model should be well established such that this application not only provides an easy way to manipulate the structural shape but also maintains a meaningful solution to the finite element based sensitivity analysis. In this environment, the designer can intuitively alter the shape of the object and examine the resulting changes to the deformation and stresses visually displayed on the virtual object which are calculated based on an approximation method. Using such a tool, a designer can easily relate shape changes to structural responses and quickly determine a optimal shape that will meet deformation or stress constraints. The image shown in Figure 1.1 illustrates the concept of the VR shape design environment. The head mounted display, or helmet, worn by the designer gives an illusion of a computer generated world surrounding the designer. The designer can perform the shape change by using the PinchGlove™ to manipulate the virtual object directly, or press the button on the virtual menu to switch

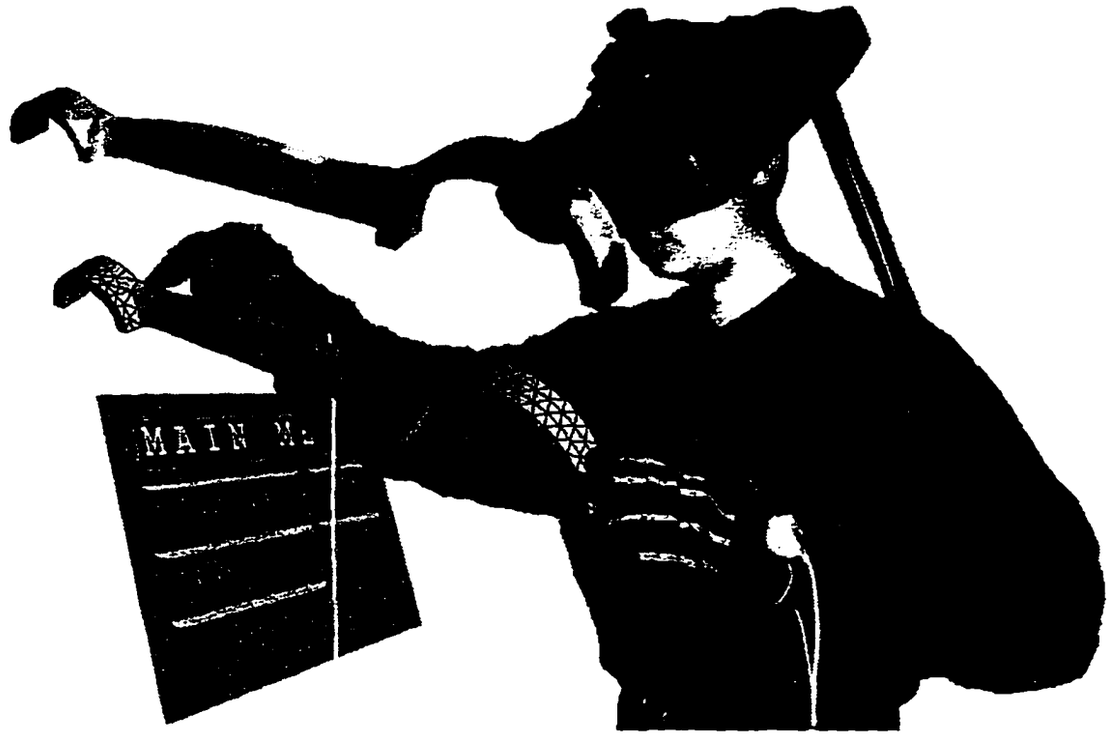


Figure 1.1: Structural shape design in a virtual environment

between different tasks. With these enhanced capabilities in visualization and manipulation, it is our belief that VR has the potential to become a powerful tool for structural shape design optimization.

1.3 Scope of the Thesis

The organization of this thesis is as follows. In Chapter 1, an overview of this project is presented. In Chapter 2, the literature reviews of virtual reality techniques applied in the field of engineering are presented. The reviews include the history of virtual reality, the introduc-

tion of general hardware and software used to build the virtual environment, and the discussion of the virtual reality technologies in academic research and industrial applications. The relative works for the techniques applied in this research are also briefly discussed.

The VR program developed in this dissertation is introduced in Chapter 3. In this chapter, the step by step procedure to perform the interactive structural shape design in the virtual environment is described. The structure of this program, which includes the hardware peripherals and software program, is also presented. The mathematical basis needed to perform the structural shape design in this project can be classified into three categories, which are: geometric shape free form deformation, sensitivity analysis, and interactive design optimization. These three categories will be discussed in Chapters 4, 5, and 6, respectively.

Chapter 4 presents NURBS-based free-form deformation (NFFD) as the tool to manipulate the geometric model in the virtual environment. Several algorithms are developed to improve the efficiency of the manipulation process. Using these algorithms, the designer will be able to perform real-time manipulation of the geometric model defined by a NURBS surface or volume.

Chapter 5 extends the NFFD technique to the deformation of finite element models. NFFD is not only used to perform the shape change of the finite element model but also to provide the derivative information of the finite element mesh in order to obtain the structural response sensitivities. In this chapter, the sensitivity computation of the NFFD model is presented. With the sensitivity information, the designer can thus perform the what-if studies of the finite element shape design in the virtual environment and determine the proper shape change based on visual feedback of the approximated analysis results in real-time.

In addition to the what-if study, another shape design task which is referred to as interactive design optimization is incorporated in this program. In Chapter 6, the mathematical form of the interactive design optimization is introduced to show the difference from the traditional design optimization formulation. This method is based on the concept that the designer can interactively change the weighting factors of the competing constraints and obtain a *designer-weighted* optimal solution.

Finally, conclusions, recommendations, and future developments of this research, are discussed in Chapter 7.

CHAPTER 2. VIRTUAL REALITY IN ENGINEERING

Virtual reality (VR) is emerging as a powerful form of computer visualization and interaction technique. It allows the user to experience a strong sense of *reality* in a computer-generated scene. Although the initiation of VR dates back to the early 1930's, it is approaching a first level of maturity in industry applications due to the low cost and high performance computer technologies in recent years. Engineers have begun to recognize the contribution of VR as an innovative tool to view and manipulate complex 3D graphic data. Because realistic images of design models provided by VR are easier to interpret and interrogate than traditional 2D drawings or 3D solid models, engineers with different backgrounds can all contribute their knowledge in the process from design to marketing. This approach holds promise to significantly simplify the design process and reduce product development time. The following sections briefly discuss the history of VR, hardware and software technologies to create a virtual environment, and VR applications in the field of engineering.

2.1 History of VR

Virtual reality can be seen as a logical evaluation of the human-machine interface. Pioneers in the field of virtual reality have been working on the media to create synthesized worlds which are not static but have interaction capability. From the human-machine interaction point of view, the Link simulator [4] can be recognized as the first ever virtual reality application to simulate a physical flight situation. The Link simulator, named after the original designer Edwin Link in 1929, was used as a pilot trainer during the World War II era. It con-

sisted of a full-scale replica of the cockpit of a plane mounted on a motion platform. The simulator used bellow-like devices to emulate the way it feels when a pilot used the controller to maneuver the airplane, and a simple horizon line that shifted served as the visual feedback. After the war, the motion platforms of training simulators for force feedback were highly developed, however, the development of visual feedback had to wait until the advent of video and digital computers.

The flight simulator provides the basic concept of virtual reality in the way that an interactive machine can be used to simulate a physical world with various situations. Morton Heilig first made this concept become reality by developing a video arcade, called *Sensorama*, in 1960 [5]. It contained stereo video, stereo sound, and a seat with handle-bars that vibrated to simulate a motorcycle ride through the streets of Brooklyn, which provided the feedback of sights, sounds, motions, and even smells. Heilig also foresaw the possibility of the head-mounted displays (HMD) and designed a mask that included 3-D slides with a wide field of view, stereo sound, and the capability to smell [6]. Unfortunately, the head-mounted display designed by Heilig never became a real product.

Ivan Sutherland was the first one to propose the feasibility to create a virtual world by the graphic display of a digital computer. The idea was to use a computer-generated display, which he called *the Ultimate Display*, to simulate the physical world such that the user can gain familiarity with concepts not realizable in the physical world by interacting with the computer-generated scene [7]. He also developed the first head mounted display (HMD) in 1968 with a mechanical tracking mechanism, called *Sword of Damocles* [8]. It consisted of two small CRTs (Cathode Ray Tubes) mounted on the helmet to display the computer-gener-

ated wire frame image onto two half-silvered mirrors placed in front of the user's eyes. The helmet was hung from the ceiling by a mechanical arm which was used for head tracking.

Frederick Brooks and researchers at the University of North Carolina (UNC) have been conducting VR applications in various fields including pharmaceutical chemistry and medical imaging since the late 1960's [4]. In the development of these applications, many VR peripherals, such as HMDs, position sensors, force-feedback systems, and advanced computer architectures have been created and made UNC become one of the world's most advanced VR research centers.

Instead of wearing a HMD, position tracker, or other bulky VR equipment, in the 1970's, Myron Krueger proposed an alternative approach to VR, call *VIDEOPLACE*. This approach projected the user's image on a large video screen where the user's silhouette could interact with the images displayed on the screen [9]. The image processing techniques were used to determine the position of the participant in order to perform the interaction with graphic objects or with other participants. He also coined the term *Artificial Reality* to convey the concept of a full-body immersion in a computer generated environment.

In the late 1970's, researchers in the Massachusetts Institute of Technology (MIT) combined the tracking device with a speech recognition system to provide an alternative interaction in the virtual environment. This project, called *Put-That-There*, allowed the user to point to a graphic object by using a magnetic transducer to track his/her hand position, and move the object through some simple vocal commands [10].

Military and other government organization such as NASA have been active in research of VR applications for training pilots and astronauts for a long time. Tom Furness worked for the Air Force to build a flight simulator called *Virtual Cockpit* in 1970's [9]. Unlike the Link

simulator using a simple line for visual feedback, the virtual cockpit used computer generated three-dimensional images as the visual feedback. However, the VR system used in the virtual cockpit was too far expensive for general applications.

In 1984, Michael McGreevy at NASA Ames built an inexpensive HMD with two LCDs (Liquid-Crystal Displays) to provide a stereoscopic image, called *Virtual Visual Environment Display* (VIVID) [11]. Scientists in NASA also used a magnetic tracker for position and orientation tracking on the VIVID instead of a heavy mechanical tracker. Later in 1985, Scott Fisher, who had participated with the *Put-That-There* project in MIT, joined NASA to continue McGreevy's works in the mid-1980's. Under his direction, the project VIEW (Virtual Interface Environment Workstation) successfully integrated a wide-angle, stereoscopic HMD, a localized sound system, and an instrumented glove in the virtual environment. Along with the development of the computer technology, the computer was powerful enough to generate flat shaded rendering of virtual objects instead of wireframe mode [11].

In the late 1970's, the instrumented glove was an important invention to provide the capability of natural interaction in the virtual environment. The first bend-sensor gloves to measure the bending of the user's fingers was invented by Dan Sandin and Richard Sayren in 1977 [9]. The data glove used in the VIEW project was developed by Tom Zimmerman and Jaron Lanier. Lanier later formed the first commercial company of VR peripherals in 1985, called VPL Research Inc., selling the sensing glove called the *DataGlove* and the head-mounted display called the *Eyephone* [11]. He is also the first one to coin the phrase *Virtual Reality* to describe an immersive, interactive, computer-generated three-dimensional environment.

Until the late 1980's, researches in VR were limited to academic, government, military, and some large companies because of the need to use expensive computer equipment. Autodesk presented an inexpensive PC-based VR system in 1989 that reduced the price in VR hardware significantly. However, the graphic speed and the quality of the rendering was limited since it served as a starter system for people who wanted to explore VR technology [9]. In the late 1980's and the early 1990's, many commercial VR companies have formed for continual evolution of VR hardware peripherals such as Virtual Research, Fakespace, Ascension, and StereoGraphics, and the development of integrated software such as Division and Sense8. Many industrial companies, such as Boeing [12][13], Caterpillar [14][15], Chrysler [12], Ford [16][17], General Motor [14][17], John Deere [18], and Northrop [19] have all begun to investigate the use of virtual reality techniques in design, training, and manufacturing simulations.

Many academic researches have also become serious about VR developments and applications. The Department of Computer Science at the University of North Carolina, Chapel Hill, led by Frederick Brooks, has been an advanced research center in VR technologies including human-machine interface, force feedback, and telepresence since 1962. Researchers at the University of Washington Human Interface Technology Laboratory (HIT Lab), under the direction of Tom Furness, have been working on the human-factor issues surrounding VR since 1989. In 1991, Carolina Cruz-Neira at the Electronic Visualization Laboratory, University of Illinois at Chicago, developed a new projection-based visualization system called *CAVE* [20]. It consists of a room where the three walls and the floor are used as projection screens to display the stereo computer generated images. Multiple users can wear shutter glasses to view the stereo images and share the same virtual environment. In the early 1990's, the National Center for Supercomputing Applications (NCSA) converted its visualization lab-

oratory into a virtual reality laboratory and used the CAVE for scientific visualization by linking the VR devices with high-performance supercomputers. The Virtual Reality Laboratory (VRL) at the University of Michigan was formed in 1993 with initial funding from the automotive industry in Detroit. VRL is focused on the industrial applications of virtual environments including virtual prototyping and virtual manufacturing processes. Iowa State University formed the Visualization Laboratory in 1986 to start research in computer graphics and began research in virtual reality in 1992. In 1996, the Synthetic Environment Laboratory at Iowa State University was formed to build an advanced CAVE system, called *C2*. The focus of research at Iowa State University is on applications of VR to industry. Other examples of research labs focused on the developments of virtual reality are Futures Lab at Argonne National Laboratory, Virtual Reality Lab at Clemson University, Synthetic Environment Laboratory at Sandia National Laboratories, VRCIM Lab at Washington State University, Biorobotics Laboratory at University of Utah, and I-CARVE Lab at University of Wisconsin-Madison.

2.2 VR Hardware and Software

Immersion is the key component in VR and serves to distinguish it from the traditional three-dimensional CAD system. Immersion refers to the user's ability to feel present in the computer generated environment. It is supported in VR by the use of a viewer-centered frame of reference and natural human motions to interact with computer models in real time. The degree of immersion experienced by users in a virtual environment depends on the quality of the visual and audio display, force feedback, and the interaction capabilities provided by the

computer hardware and software. The hardware and software technologies to build a virtual environment can be categorized as *Visual Displays*, *Position Tracking Devices*, *Computation Hardware and Software*, *Haptic Devices*, *Audio Systems* and *Interface Devices*. The following sections will briefly introduce each category. More information for each category can be found in references [1],[2],[3],[21], and [22].

2.2.1. Visual Displays

Visual displays are the devices used to present the computer image to the user. Depending on the degree of immersion, different kinds of visual displays are currently available including traditional computer monitors, head mounted displays (HMDs), head-coupled displays (HCDs), and projection systems. Examples of HMDs include FS5 from Virtual Research and Datavisor from n-Vision. Examples of HCD include BOOM, FS, and PUSH systems from Fakespace. Examples of projection systems include CAVE [20], Fakespace's Workbench, and Pyramid's ImmersaDesk.

The traditional computer monitor coupled with stereo glasses can be considered as the most basic and least immersive visual display. This kind of visual display, sometimes referred to as *fish tank display*, provides inexpensive, convenient, and high resolution set up for VR visual feedback. However, the visual display can cause the stereo frame violation to detract from the degree of immersion due to the fact that the physical monitor frame can be seen with the stereo image. The HMDs and HCDs are common VR peripherals for higher immersive visual displays. These devices use small CRTs (Cathode-Ray Tube) or LCDs (Liquid Crystal Display) placed in front of user's eyes which provide a wider field of view (FOV) through the use of LEEP optics to magnify the image. The LEEP (Large Expanse Extra Perspective) wide-

angle viewing optics was developed by LEEP System/POP-Optix Labs to avoid the distortion by using a wide-angle, or *fish-eye*, lens [1]. This has become a standard in most HMDs and HCDs. The other visual display paradigm in VR is the use of projection technology. The projection systems provide the widest FOV and high resolution by physically surrounding the viewer with the computer generated image projected on the screen. This eliminates the need for using HMDs or HCDs to create a wider and more natural FOV, and multiple users can share the same virtual environment due to the nature of the surrounding projection screens.

Note that all of the above systems are capable of stereoscopic views to provide the user with a higher degree of immersion. The stereoscopic cues resulting from binocular disparity or stereopsis conveys important information for the viewer to understand the depth perception of a spatial object. In the virtual environment, the stereopsis is obtained by creating two images from the viewpoints of two eyes and blending the images to create the stereo illusion. In HMD and HCD visual displays, these two images can be displayed simultaneously through two different graphic channels for two eyes. In the computer monitor and projection system visual displays, the two images are displayed alternatively. Through the use of LCD shutter glasses which synchronize with the computer to display the different image for each eye at a high flashing rate. Also the images can be displayed simultaneously through polarization technologies such that each eye can see the corresponding image through polarized glasses [11].

2.2.2. Position Tracking Devices

In the virtual environment, natural human motions are used to control the viewpoint and the manipulator. A tracking device is thus necessary to measure the position and orientation of the sensor mapping to the user's head or hand position and orientation in the virtual environ-

ment. Based on the techniques used to track the sensor, the most popular tracking devices can be categorized as magnetic trackers, acoustic trackers, and mechanical trackers. The magnetic trackers, such as the Ascension Flock of Birds and Polhemus Fastrak, use a transmitter that emits an electromagnetic field which is detected by the receiver to determine the spatial position of the receiver. The acoustic trackers incorporated in the Logitech 6D Mouse and Mattel PowerGlove, use at least three microphones on the receiver to receive ultrasonic pulses from the emitter. The position and orientation of the tracker are determined by the differences in the pulses reaching each microphone. The above two tracking methods are relatively inexpensive and have a wide range of use but suffer from latency and environmental interference. The mechanical trackers, such as those used with the Fakespace BOOM and Immersion Probe-MD, are rigid mechanisms composed of linkages and joints which measure the position and orientation of the end effector. Although they can be relatively accurate without environmental interference and have little intrinsic latencies, the workspace is limited by the range of the mechanical linkage.

2.2.3. Computer Hardware and Software

Computer hardware and software are used to develop and control the overall operation of the virtual environment. The computer hardware refers to a graphic workstation which is capable of real-time, three-dimensional graphic display at high frame rates, integrating and synchronizing the input and output of the hardware peripherals, and even a fast communication capability through the network. The software system includes hardware drivers, interaction, navigation, and graphic rendering software. The hardware drivers are used to initialize the hardware, such as data gloves and position trackers, and take raw inputs from the hardware

then interprets the input data. Interaction software provides a mechanism to map the user's actions, such as moving a virtual object by hand, and physical properties, such as collision detection, to the environment. Navigation software provides a mean to move through the virtual environment. It includes the determination of the position and orientation transformation matrix for the current scene of the virtual environment as the user changes the viewpoint. Some polygon reduction and level-of-detail techniques are also important in the software for real-time navigation in the virtual environment. The graphic rendering is used to construct the virtual environment on the visual display device. It includes the techniques to draw virtual objects, with lighting, material properties, and texture effects to make the object look like the physical model in the real world.

2.2.4. Haptic Devices

Haptic devices are used for manual exploration and manipulation of virtual objects. They allow the user to *feel* the virtual object by applying force and tactile feedback from the hardware devices. Haptic hardware devices can be classified into three major types: ground-based devices, body-based devices, and tactile sensory devices. The ground based haptic devices are fixed on the ground, such as joysticks and robot arms, to give the force reflection to the user's hand through the handle of such devices. The body-based devices, such as a flexible exoskeleton that can be worn by the user, or the rigid exoskeleton that is composed of jointed linkages affixed to the user, are designed to fit over and move with the limbs or fingers of the user. The above two devices will give the net force feedback for coarse object interaction in the virtual environment. However, tactile information is sometimes needed in order that the user can feel the texture or slip of the virtual object. The tactile sensory devices display

tactile information through vibrating pins (Vibrotactile) or electrodes (Electrotactile) to stimulate simple textures and physical features of the virtual object. Other haptic devices use pneumatic techniques to simulate the feeling of touch by inflating or deflating the micro air pockets on the glove as the glove touches a virtual object.

2.2.5. Audio Systems

Sound can be considered as a supplement of visual information to enhance the overall perception for the user's interaction in the virtual environment. The audio system has been well developed such that the sound can be created, edited, and played back through digital synthesis. In VR, the computer-generated sound can simulate the spatial nature of the environment such that the sound can appear from a particular place in the virtual environment. The spatial and directional perception of sound, refers to *sound localization*, are primarily based on interaural time differences and interaural intensity differences of the sound wave reaching into both ears from the sound source [1]. Interaural time differences occur when a sound source is closer to one ear than the other thus the sound wave arrives at each ear at slightly different times. Interaural intensity differences are based on the fact that the sound intensity drops off with distance therefore the intensity of the sound arriving at each ear is different. These interaural time and intensity differences of sound are important for people with biaural hearing to distinguish the spatial position of the sound source. The other factor to localize a sound source is called *acoustic shadow* [1] due to the fact that the high-frequency sounds are easily blocked by an obstacle. It is useful in the virtual environment to provide the information of a relative position of the sound source and the blocking virtual object.

Instead of two channels of speakers that create a two dimensional sound, the audio system in the virtual environment should be able to provide sound localization and interaction. Sound localization depends on the factors described above. However, as the user turns his/her head in the virtual environment, the interaural time differences, interaural intensity differences, and acoustic shadow of the sound source should also be different depending on the position and orientation of the user's head in order to localize the sound source. Thus the sound interaction should couple with the position tracker in order to determine the intensity and synchronization of all sound channels. Sound interaction also includes the audio feedback as the user interacts with virtual objects. For example different synthesized sounds or prerecorded sounds can be used as an aid for contact detection as the user touches different virtual objects.

2.2.6. Interface Devices

The interaction in the virtual environment is not limited to navigation by using position tracking devices. In addition to navigation, interface devices are used to pass commands and perform manipulation in the virtual environment. Data gloves can be used to pass gesture commands and interact with virtual objects. These devices are often used in conjunction with the position tracking devices to tracking the position of the manipulator. Some body-based linkage interface can be used to capture the human body locomotion or the expression of the human face. Other joint-based devices, such as joystick and spaceball, can also be used in tracking the viewpoint or moving a virtual object like position tracking devices. The voice command input device can provide a simple and easy way to pass commands into the virtual environment through a voice recognition system. Proper training of the voice-recognition sys-

tem should be done beforehand such that the system can capture the voice patterns for a variety of vocal commands.

Virtual reality techniques have been around for decades. Prior to the early stage, real-time interactive VR applications were targeted mainly to expensive flight simulators [23]. With ever increasing computer capabilities and improving VR technologies, many researchers have been successfully applying VR in various fields including medical applications, architectural walk-through, entertainment, education, science and engineering. In the next section, the current usage of virtual reality techniques in the field of engineering will be briefly discussed.

2.3 Engineering Applications of VR

The following sections introduce VR applications in engineering including design visualization and verification, prototyping and conceptual design, manufacturing and assembly planning, concurrent engineering, telepresence, education and training. For each category, the role of VR in the improvement of current technology is emphasized, and the significant advances in research and industrial applications are cited. Note that these categories are not mutually exclusive and some research and applications may overlap.

2.3.1 Design Visualization and Verification

The advanced visualization and interactive capabilities of VR have made it a natural evolution of existing tools for engineering design visualization and verification. Studies [24] have shown that the ego-referenced frame provided by VR is more effective to relay the location and its spatial relationship in the design space than the world-referenced frame of the traditional computer interface. Due to this advantage, most engineering applications are using

VR as a tool to visualize complex geometric models imported from CAD packages or the mathematical data of the analysis results for design verification.

The NASA Ames Research Center developed a virtual environment, called *Virtual Windtunnel*, to visualize a simulated flow field which is not easy to interpret through normal numerical simulations [1]. A CAD model of a space shuttle and the turbulent flow-field information around the shuttle are brought into the virtual environment. The researchers can analyze the effect of the turbulent flow-field visually as they were actually standing inside a wind tunnel. The Virtual Reality Laboratory in Ford Motor Corporation has been using a BOOM to visualize the airflow patterns in and around vehicles [16][17] to help engineers analyze the effects of engine component cooling properties. At the London Motor Show in 1993, Volvo demonstrated coupling a VR environment with finite element analysis results and a HMD to view a side-impact car crash from the viewpoint at the driver's seat [14]. General Motor also used a CAVE to view a full scale car crash simulation [25].

2.3.2 Prototyping and Conceptual Design

Most engineering applications of VR today are focused on the development of virtual prototyping and conceptual design. Since the conceptual design and prototyping in the initial design stage is a repeated process to continuously refine the design, it is inherently time-consuming and expensive. With photorealistic texture mapping, stereoscopic view, and even full scale images, the virtual prototype can be as real as a physical one. If the refinement of the prototypes can be performed in the virtual environment, a significant efficiency improvement can be achieved by reducing, even eliminating, the need for rebuilding costly physical mock-ups [12].

Engineers in Ford Motor Company has been using virtual prototypes in the vehicle design to investigate the accessibility and the assembly process of engine parts [17]. The intent is to reduce the number of physical prototypes that must be built to refine the design model. Chrysler Corporation engineers were cooperating with the Virtual Reality Laboratory at the University of Michigan to evaluate the process of virtual prototyping of automotive interiors. The prototypes of a car's interior is used to study the visibility, reachability, aesthetics and ergonomics [12][26]. This application is using a BOOM and a DataGlove for the virtual model viewing and interacting. General Motors (GM) was also using virtual prototyping to evaluate vehicle interior design. GM was the first automotive corporation to build a CAVE system for virtual prototyping [25]. The CAVE system creates a highly immersive environment which can provide a full-size stereo image with deep and correct perspective of the virtual prototypes. Engineers in different fields can easily understand the content of the virtual prototype and use the system as a communication tool to contribute their expertise. Researchers at Caterpillar Inc. were working in collaboration with the National Center for Supercomputing Applications (NCSA) to perform visualization studies on virtual prototypes of heavy equipment through the use of the CAVE system [12]. The system allows the designer to sit in the driver's seat of the design vehicle and look around to evaluate the degree of visibility. NCSA also works with Germany's National Research Center for Information Technology (GMD) to develop a collaborative virtual prototyping system for Caterpillar, called VPS, such that the engineers in the US and in Germany can work together using distributed virtual reality technology. The system supports the interactive redesign and real-time communication of the design teams in a shared virtual environment at each remote site's viewpoint [15].

Virtual conceptual design intends to create the design model or refine the model all in the virtual environment without external CAD software. The I-CARVE Laboratory at the University of Wisconsin-Madison has developed a virtual conceptual design system, called COVIRDS [27][28]. The system allows the designer to create a virtual object and add features on the object by using natural hand motions, gestures, and even vocal commands. Researchers at the University of Illinois developed a VR-based CAD system [29] for the designer to specify features of a solid model in a CAVE environment. Researchers at Iowa State University were working on the VR applications in spherical mechanism design [30]. This program allows the user to specify the path of the coupler link and the positions of the fixed and moving pivots by reaching into the virtual world using the CyberGlove. The 3D position specification is not easy to implement by using a mouse on a 2D computer monitor, however, it is easy and intuitive to perform such a task in the virtual environment.

2.3.3 Manufacturing and Assembly

Virtual manufacturing and virtual assembly planning refers to the manufacturing and assembly process performed in the virtual environment before committing to physical manufacturing production or assembly line. Thus the manufacturing and assembly process can be executed and evaluated beforehand without the use of actual equipment and materials. Difficulties due to the manufacturing and assembly tasks can be discovered in the design process and thus costly redesign can be avoided and safety issues can be identified early in the design process.

The *Virtual Manufacturing Environment* presented by Jayaram and Myklebust allows the designer to use virtual manufacturing facilities during the design process and actually per-

form the design by manufacturing the part from the raw materials [31][32]. Jayaram *et al.* also developed an integrated virtual environment to simulate the assembly and manufacturing process [33]. Massachusetts Institute of Technology has also developed the *Virtual Environment for Design for Assembly.* This virtual environment holds the potential to provide the designer with early assembly information such as ease of handling and ease of assembly [34]. Researchers at Boeing are developing *Augmented Reality* (AR) to guide the technician in assembling a wiring harness that forms part of an airplane's electrical system [13]. AR is another virtual reality technique where computer-generated displays are superimposed on the real environment by using a see-through HMD. The *Automated Airframe Assembly Program* (AAAP) developed in Northrop is a virtual environment for assembly modeling simulation that allows the designer to manipulate the parts to verify tooling and assembly strategies for building airplane assemblies [19].

2.3.4 Concurrent Engineering

Virtual reality provides advanced visualization capabilities that enables people from all backgrounds to easily and accurately understand the form and function of the design model. These capabilities become even more crucial in concurrent engineering (CE) where each member of the design team in different fields of expertise can contribute to the design process by bringing different knowledge and skills to bear on solving the design problem. To achieve the involvement of all members in a design team, the 3D design prototype, with full-scale and photorealistic stereo image provided by VR, can be a common communication tool for the trade-off design from the views of manufacturability, economics, maintainability, and even marketing of all CE team members.

General Motors Research and Development Center has developed a CAVE-based visualization system, called *VisualEyes*. The enhanced visualization of the design prototype provided by the CAVE has solved some fundamental communication problems in the concurrent engineering design team, which includes people in the fields of design, engineering, and manufacturing [25]. Designers and engineers at Boeing used VR to help the design team of Boeing's 777 project to evaluate the virtual model in order to improve the efficiency in the cooperation of the team members and to avoid cost overruns and delays [16].

2.3.5. Telepresence

Telepresence refers to a computer-generated environment to simulate a real but remote environment such that the user can feel the illusion of presence in that remote location and perform different kinds of tasks remotely. The typical applications of telepresence include teleconference and telerobotics [1].

VR teleconferencing combines the VR techniques and fast network communication such that the participants in different locations can share the same virtual environment through telecommunication lines. Unlike the videophone conference such that only two dimensional images of the conference attendees are shown on the computer monitor, VR telepresence generates a three-dimensional virtual conference environment such that all attendees can feel present in the same conference room and can interact with each other by using a data glove or other interaction devices. VR telepresence is currently under development in many research groups and the major obstacle right now is the limitation for conveying large video information through a network in real time.

Telerobotics combines the techniques of VR and robotics such that the user can operate a robot at a remote site from a local virtual environment, and the remote robot provides feedback of the visual, audio, and haptic information to the user so that the user can feel present at the remote site and perform tasks in that remote area. The applications of telerobotics are mainly focused on the operations in some hazard areas such as battle fields or nuclear, toxic pollution zone, and some environments, such as deep ocean and outer space, in which human beings were not meant to dwell. Back to 1976, P. J. Kilpatrick at the University of North Carolina developed simple telerobotics that combined a radioisotope manipulator with a simple graphic environment [9]. The system, called GROPE-II, used a mechanical arm to operate a graphic manipulator in a virtual environment with a stereo visual display. The mechanical arm provides a net force feedback if the graphic manipulator touches a virtual object. Researchers at the University of Utah developed a teleoperating system with an anthropomorphic (human-like) robot arm unit [35]. The system contains of an exoskeleton master arm worn by the user, and the remote slave robot arm performs a kinematically identical movement as the master arm operated by the human operator. Another fully anthropomorphic telerobot, called the *Green Man*, was built at the Naval Oceans Systems Center (NOSC) laboratory [1]. The system consists of a slave robot with two arms, cameras and microphones to provide force, visual, and sound feedback of the remote environment respectively, and a master hinged mechanical exoskeleton equipped with mechanical tracking sensors and a CRT-based HMD with earphones to control the remote robot and show the images and sounds fed from the robot.

2.3.6 Engineering Training and Education

VR can provide a simulated environment to represent a real situation in such a way that training and education can be achieved outside the real situation. VR has long been used to train pilots back to the day of the LINK simulator in War World II. Along with the improvements in the computer technologies, the training and education in many fields, from scientific education to complicated military mission, have been using VR to take advantage of its high immersion, low risk and cost efficiency.

NASA's Johnson Space Center has been using VR in training astronauts for maintenance missions in space [36]. The objective of the virtual environment is to allow the trainee to control the movement in space by using two hand-controllers to simulate the Manned Maneuvering Unit (MMU), and replace the faulty battery in the Hubble Space Telescope with a new battery using a dataglove interacting with the virtual model of the Hubble Telescope. The Pentagon's Advanced Research Project Agency (ARPA) has developed a VR combat simulator to train soldiers in a high-immersive, low-risk battle [19]. The simulator can generate the feel of a tank in the battle field and the soldier can maneuver the tank and shoot the enemy using a joystick. Sandia National Laboratory is developing a VR system to train people to handle a specific situation or set of scenarios. The system, called *situational training*, allows the user to wear a HMD and a data glove to control a graphical representation of the user, called *avatar*, in the virtual environment to train for some dangerous situations such as the training of a bomb squad in searching techniques [37]. Researchers at Iowa State University are working on the development of a virtual environment for training manufacturing technicians [2]. The virtual environment consists of various manufacturing equipment such as robots and NC

machines. Workers can be trained to use the equipment by interacting with a virtual control panel the same way they interact with the physical panel.

2.4 Relative Works of this Research

Although VR has recently approached its first step of maturity in the field of engineering applications, there is very little documented research in the applications of coupling VR techniques with finite element analysis and structural shape design. Yagawa *et al.* [38] developed a virtual reality application which integrates the constructive solid geometry (CSG), automesh techniques, and finite element post-processing. The designer can use shutter glasses to view a stereo image of a finite element model, and use the Logitech 3D mouse to interact with a three-dimensional menu to perform different pre and post-processes. This application is essentially the same as traditional finite element pre and post processors but with additional stereo viewing and 3D interaction capability on a desktop workstation. Yeh and Vance [39][40] incorporated finite element data into a virtual environment such that the designer can interactively modify the transverse force using an instrumented glove and see the resulting deformation and stress contour of a structural system in a stereo view. The structural shape doesn't change in this application, however, the concept of using the sensitivity information to interact with a finite element model in real-time has been adopted in this research. Haase and Preβ [41] proposed an interaction techniques to modify the finite element model and apply boundary conditions for the finite element pre-processing. A linear function is used to modify the finite element node and its neighboring nodes, thus a smooth shape of the finite element model can not be achieved.

In order to obtain a smooth shape of the finite element model after modification, NURBS-based free-form deformation (NFFD), proposed by Lamousin and Waggenspack [42], is applied. The NFFD is original designed to modify the free form shape of a geometric model. In order to maintain a meaningful finite element shape after deformation, additional considerations are required such as maintaining the linearity of the interior finite element mesh and preserving the features of the finite element model.

The interactive design optimization applied in this research follows the concept of the designer-weighted cost function presented by Rizai and Bernard [43]. It is incorporated in this research in order to take advantage of the fast convergence of this optimization method to facilitate the real-time interaction in the virtual environment. In addition this method allows designers to interactively vary the relative importance of each design constraint of the problem.

A general purpose VR program is developed as a result of this research. The program combines virtual reality techniques, sensitivity methods, design optimization, and free-form deformation to facilitate engineering design optimization. The next chapter will outline the structure of this program for structural shape design in a virtual environment.

CHAPTER 3. INTEGRATED VIRTUAL ENVIRONMENT FOR STRUCTURAL SHAPE DESIGN OPTIMIZATION

The software program developed in this research combines the virtual reality techniques, free-form deformation, finite element analysis, sensitivity analysis, and interactive design optimization to facilitate the finite element based structural shape design optimization. In this chapter, procedures to perform the interactive structural shape design in the virtual environment are presented, and the program structure of the integrated virtual environment is also introduced. An engine connecting rod design is used as an example to illustrate the design steps of the interactive structural shape design in the virtual environment. The software to build the virtual environment is Sense8's WorldToolKit (WTK) [44]. It consists of a set of C functions which create the graphical virtual environment and provide hardware drivers for managing the peripherals used for navigating in the virtual world and interacting with virtual objects. The program is written in the C^{++} programming language to take advantage of its object-orientation capability to construct the overall program structure. The finite element model is used in this project for the structural shape design because it can not only be considered as a secondary representation of the geometry model, but also provide the analysis capability of the structural responses such as displacements, stresses, and mode shapes. The MSC/NASTRAN [45] data format is used as the file input/output protocol since it contains explicit information about the finite element mesh and analysis results in ASCII form, and the designer can take advantage of the finite element solutions of MSC/NASTRAN to obtain the sensitivities of the structural responses and the interior mesh movements for the interactive design. Furthermore, most commercial finite element preprocessors with automatic mesh gen-

erators can provide the finite element mesh output in MSC/NASTRAN bulk data file format thus the VR program developed in this project can readily accept the design model from most CAD packages.

3.1 Procedure for the Interactive Structural Shape Design in VR

The procedure to perform the interactive structural shape design in the virtual environment is shown in the flow chart of Figure 3.1. It can be generally categorized into three major

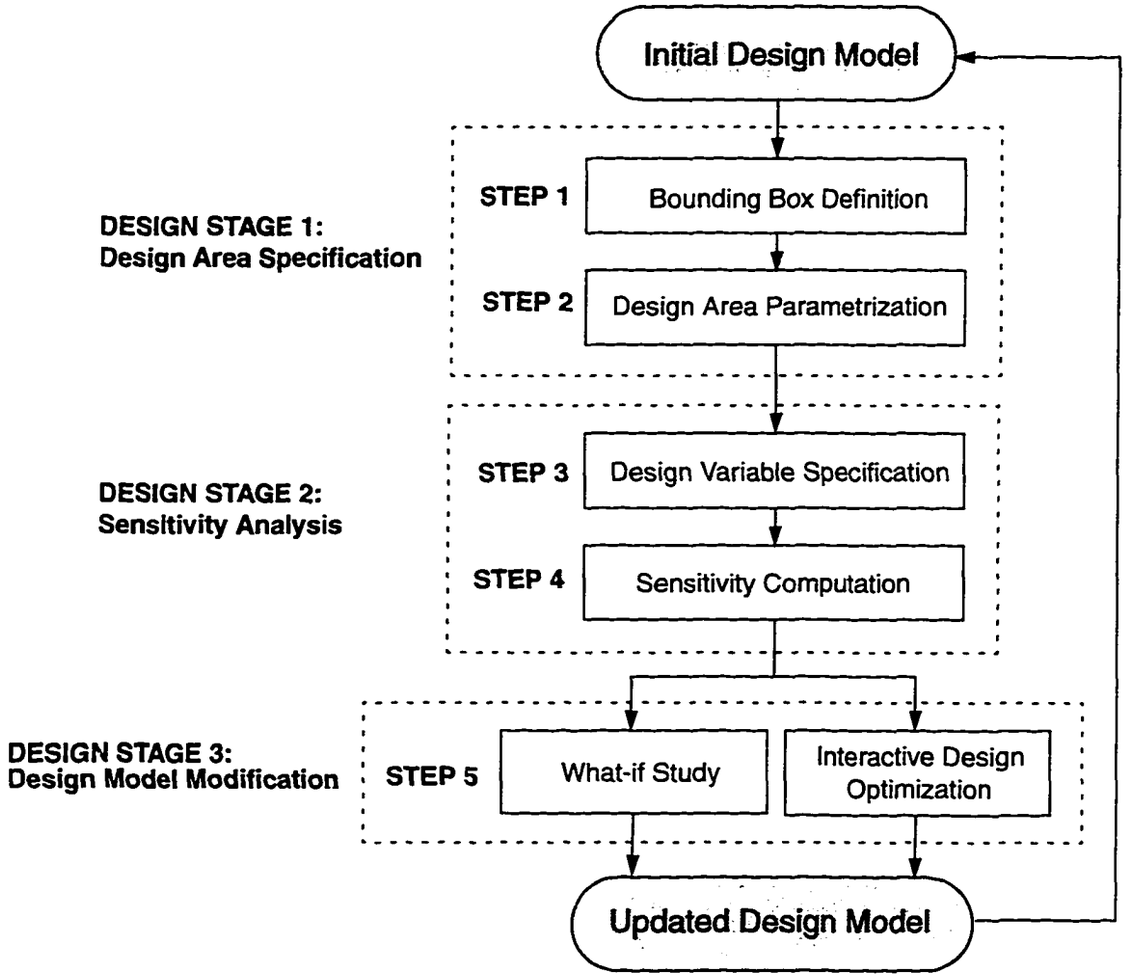


Figure 3.1: Procedure of the interactive structural shape design in VR

design stages which are *Design Area Specification*, *Sensitivity Analysis*, and *Design Model Manipulation*. Within these process there are five design steps which include *Bounding Box Definition*, *Design Area Parametrization*, *Design Variable Specification*, *Sensitivity Computation*, *What-if Study*, and *Interactive Design Optimization*. The following sections will discuss these design stages and the design steps in detail.

3.1.1 Design Area Specification

To perform structural shape design, the designer should first specify the bounding box which defines the design area to be modified. The bounding box is the control point lattice which constructs the NURBS surface or volume for the deformation of a 2D or 3D finite element model respectively. The finite element model can be fully embedded inside the bounding box for global modification, or partially embedded for local modification depending on the designer's knowledge of the design model.

The first design step, *Bounding Box Definition*, is to manipulate the bounding box in order to cover the desired design area, and to define the knot vectors as well as the control point coordinates for a NURBS surface or volume construction. The initial design of the finite element model and its analysis results are loaded into the virtual environment and displayed as a virtual design object and an analysis result object, respectively. As shown in Figure 3.2 (a), a quarter model of the engine connecting rod is first designed by using SDRC/I-DEAS [46] and automatically meshed using tetrahedron solid elements. Static analysis is performed using MSC/NASTRAN and the results of displacements and vonMises stress are stored as NASTRAN punch file format. A virtual menu, which serves as an aid for the designer to specify different tasks, can be opened or closed using simple hand gestures. By pressing the button

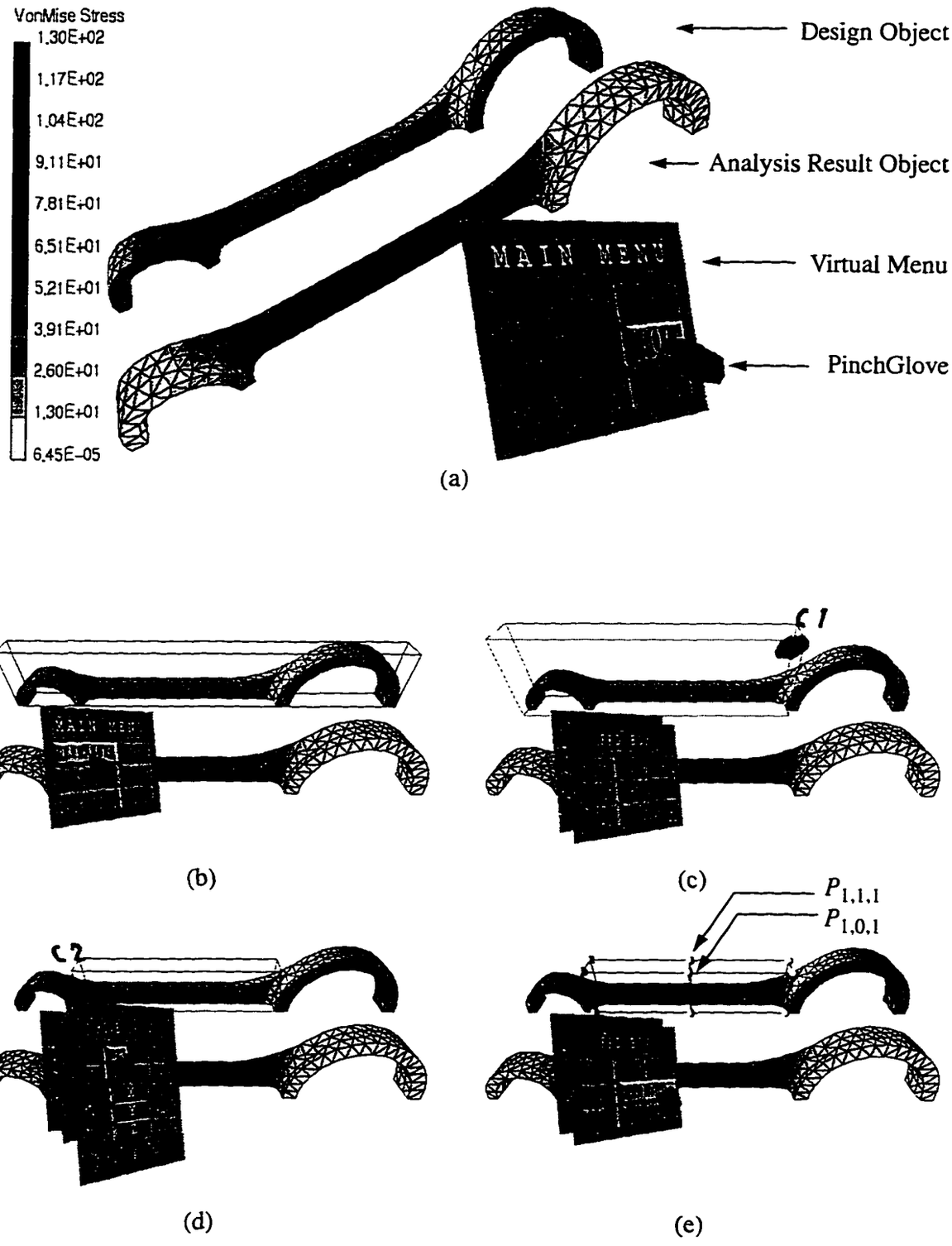


Figure 3.2: The design steps in the *Design Area Specification* stage

FFD BOX on the virtual menu, a default bounding box is shown and covers the entire design object as in Figure 3.2 (b). The designer then can use the picking device, for example the PinchGlove™ in this case, to specify the design area by grasping the corner of the bounding box to resize it (Figure 3.2 (c)), or use the menu system to change the size using precise movements to cover the desired section of the connecting rod (Figure 3.2 (d)). After the desired section is specified, the parametrization process is executed by pressing the button **OK**.

The second design step, *Design Area Parametrization*, is a process to embed the design area of the finite element model inside a NURBS surface or a volume defined by the control point lattice of the bounding box. This allows the user to change the NURBS surface or volume, and obtain the resultant changes to the finite element model. The control point lattice is defined in a configuration file which allows the designer to specify the locations of the control points. The specification of the configuration file will be introduced in Appendix A. From the mathematical point of view, the parametrization process in NURBS is to use the inverse mapping technique to find the parametric value of the finite element nodal coordinates inside the bounding NURBS surface or volume. Detailed information of the inverse mapping technique will be discussed in section 4.1.2. The design area will be highlighted to indicate that the parametrization process is completed (Figure 3.2 (e)), and the design process is ready to proceed to the next design stage.

3.1.2 Sensitivity Analysis

In the third design step, the designer specifies the design variable which controls the shape change of the specified design area. The design variables in this project can be either the control points on the bounding box, or the finite element node on the design model. The

design variable will be highlighted once it is specified by using the PinchGlove to pick the desired control points or the finite element node. In the connecting rod example, the control points $P_{1,0,1}$ and $P_{1,1,1}$, as shown in Figure 3.2 (e), are specified as the design variable. Since these two control points are moved identically, they can be considered as one design variable. This is generally referred to as *reduced basis method* which will be discussed in Section 5.4.4. After the design variable specification, the fourth design step, *Sensitivity Computation*, is performed by pressing the button **SEN ANS**. A MSC/NASTRAN data file will be generated and the analysis will be executed automatically to compute the sensitivities of the structural

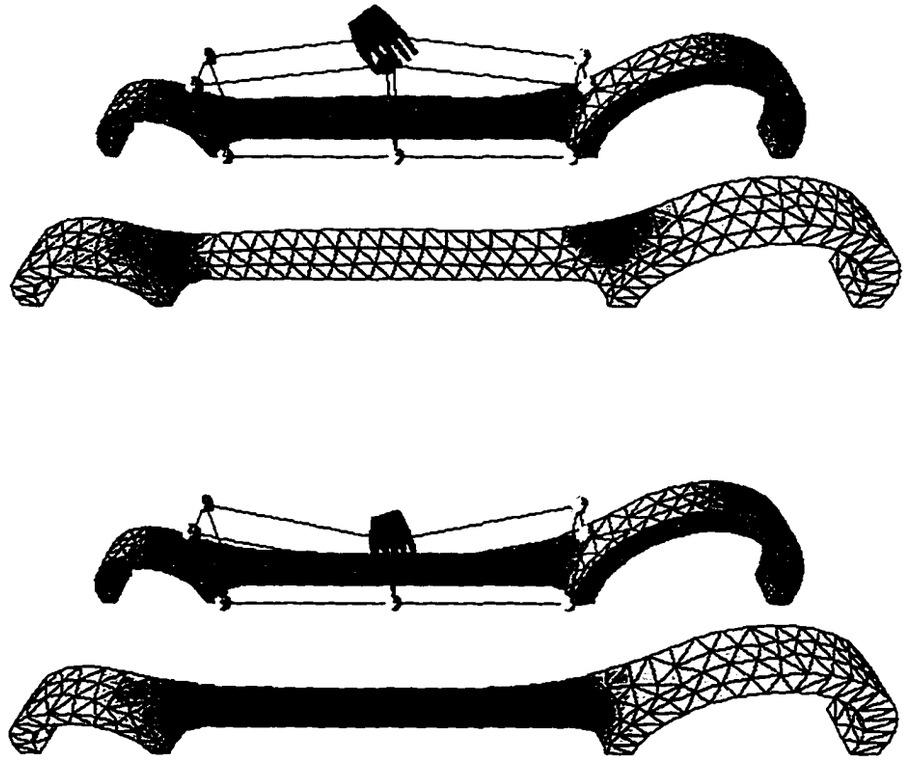


Figure 3.3: Shape changes of the engine connecting rod and the corresponding approximated analysis results

responses and the sensitivity of the mesh movements, or design velocity field. The sensitivity information will be used for the what-if studies and the interactive design optimization in the next design stage to display the approximated structural responses of the modified design. The designer is allowed to choose multiple design variables, however, the MSC/NASTRAN analysis should perform separately for each design variable. Detailed information about the sensitivity computation will be discussed in Chapter 5.

3.1.3. Design Model Manipulation

Once the sensitivity information of the structural responses and the design velocity field are obtained, the designer can choose either the *What-if Study* or the *Interactive Design Optimization* to perform the structural shape design task. The purpose of this design step is to guide the geometry shape design with structural analysis consideration, thus the designer will gain the visual feedback of the effects that a geometry shape change has on the analysis results, and obtain an optimal design in a timely manner. In the *What-if Study*, the designer can use the picking device to grasp the design variable, as shown in Figure 3.3, and change the shape of the design object. With the sensitivity information, the analysis result object can be updated correspondingly using a linear approximation of the displacement and stress in real time. For the *Interactive Design Optimization* task, the designer should first press the button **DES OPT** on the main menu. A menu with sliders, as shown in Figure 3.4, will pop up for the designer to modify the design problem by changing the weights of the design parameters. The optimal shape of the specified design problem will be obtained after the designer presses the button **APPLY**. Detailed information of these two tasks will be discussed in Chapter 5 and 6 respectively.

3.2 Architecture of the Program

In this section, the software structure to construct the program is introduced, and the hardware peripherals used in the virtual environment are also discussed. As shown in Figure 3.5, the architecture of the program includes several global modules: *FEA Data Base*, *Virtual Environment*, *Free-Form Deformation*, and *Sensitivity Analysis*. From an object orientated data structure point of view, the global module is essentially a C^{++} class or hierarchic classes. Each global module contains several local modules which perform independent tasks but share the same private data of the global module. Next the functionality of each module will be discussed.

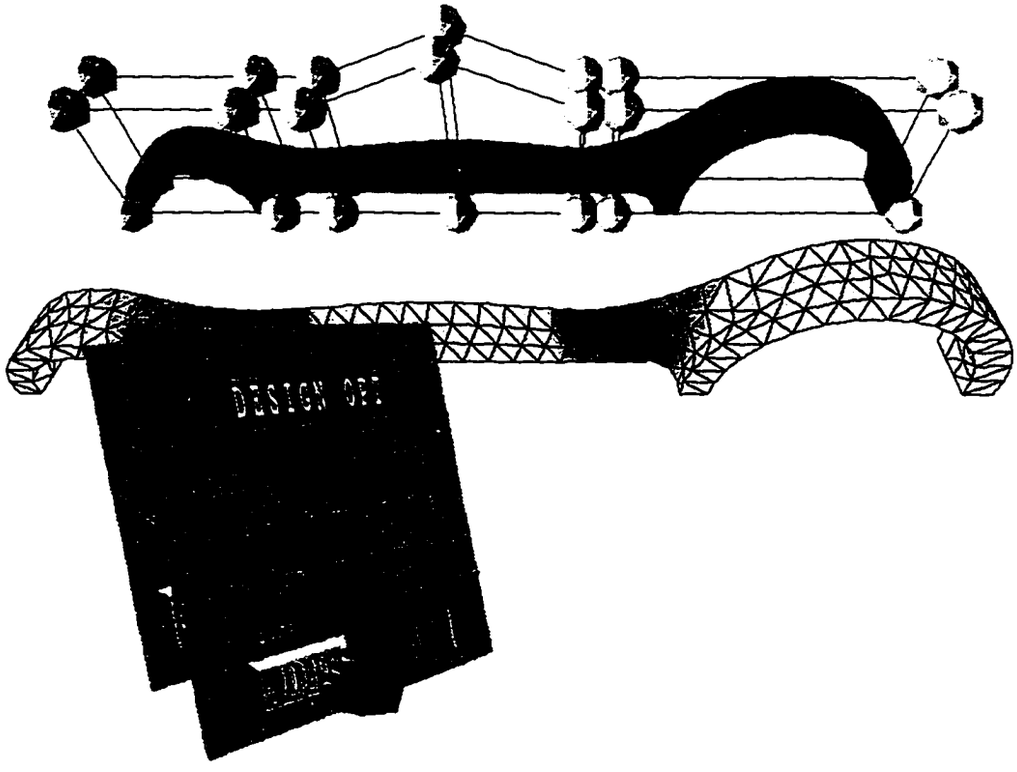


Figure 3.4: Menu for Interactive Design Optimization Task

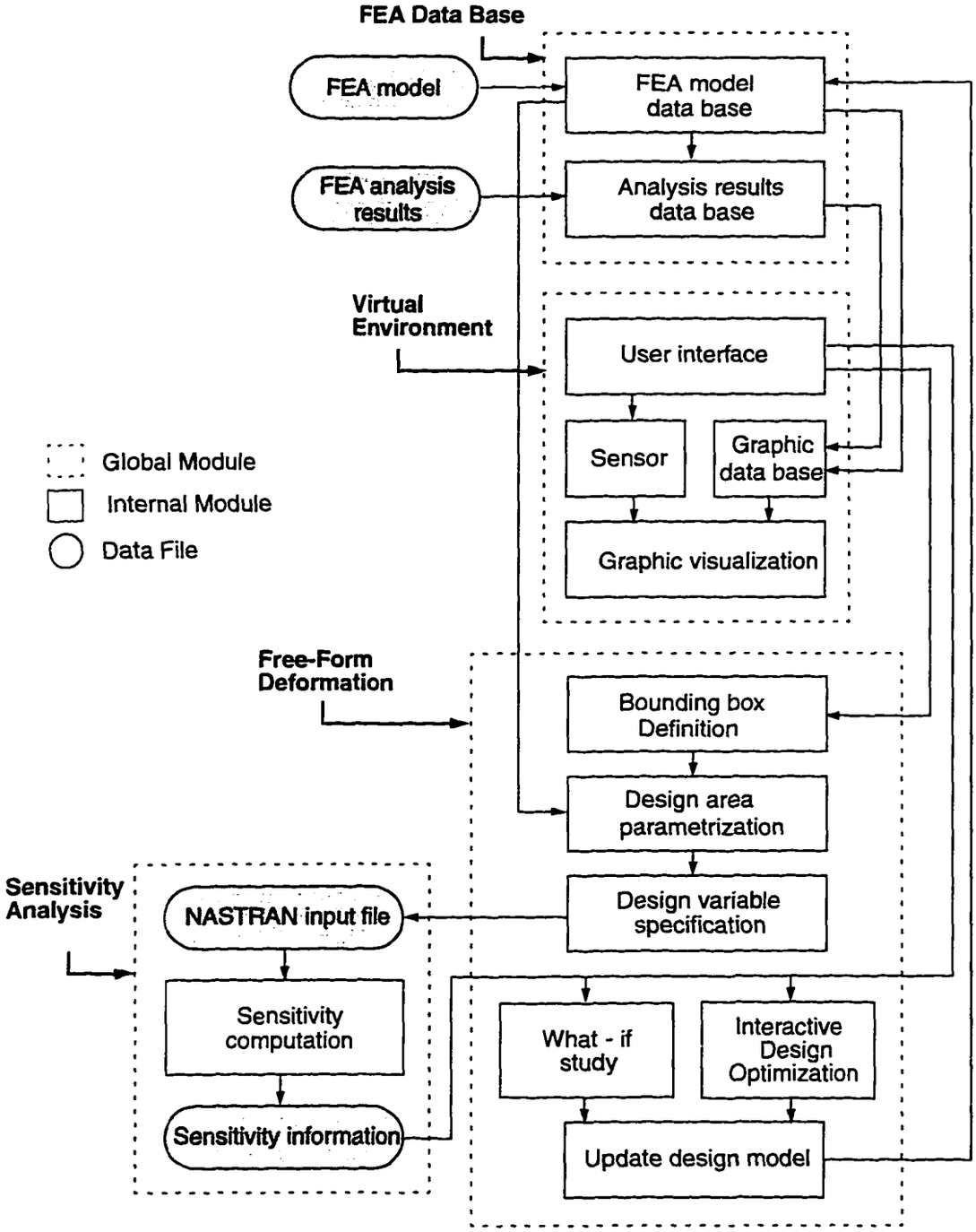


Figure 3.5: Program architecture

3.2.1 FEM Data Base

The *FEM Data Base* contains two internal modules to read the data files of the finite element mesh (*FEA model data base*) and analysis results (*Analysis results data base*). The finite element mesh information can be obtained from any CAD package which can output MSC/NASTRAN bulk data format. The bulk data format contains the grid point coordinates and the connectivity of the elements which is essential for displaying the graphic objects in the virtual environment. The boundary nodes and the interior nodes will be distinguished in this module in order to perform the interior mesh movement according to the movement of the boundary nodes. The solid element will also be divided into several polygons in order to increase the rendering efficiency since only the boundary polygons are visible for displaying the graphic object. The acceptable element types of the finite element model in this program are listed in Table 3.1. More element types can be easily incorporated into this module depending on the user's need.

The analysis results are taken from a MSC/NASTRAN punch file which contains the grid point displacement and vonMises stress information. In order to display the color contours on the analysis model, the stress value at each nodal point must be found. For some

Table 3.1: Acceptable element types of the finite element model

NASTRAN ELEMENT	ELEMENT TYPE	NUMBER OF NODES
CTRIA3	triangle shell element	3
CTRIA6	triangle shell element	6
CQUAD4	quadrilateral shell element	4
CQUAD8	quadrilateral shell element	8
CTETRA	tetrahedron solid element	4 - 10
CHEXA	hexahedron solid element	8 - 20

element types in MSC/NASTRAN only the element vonMises stresses can be output to the punch file. The vonMises stress at the nodal point is then determined by the average of the neighboring element vonMises stresses. The stress contour data is then generated in this module through the linear interpolation of the stresses of the adjacent grids [47] to show the color contour on the deformed model.

Once the data file of the finite element mesh and the corresponding analysis results have been loaded into the program, the virtual environment is brought up for the designer to view the design model through advanced VR technology and to perform the free-form shape design and investigate the effect that the design will have on the analysis results.

3.2.2 Virtual Environment

The virtual environment module contains the program to control the VR peripherals and display the virtual image for the interactive structural shape design. The software structure of this global module includes *user interface*, *sensor*, *graphic data base* and *graphic visualization*.

The *user interface* module contains the tools that a designer needs to interact with the virtual world. It includes the capability for collision detection of the data glove with the virtual object to indicate that the designer is *touching* the object, and hand gesture interpretation to perform different tasks such as picking and moving an object. However, the desired interaction in the virtual world is more complex than just manipulating a virtual object directly through the use of the data glove. For operations that may be difficult to specify using hand gestures, a more menu-driven approach is supplied. The incorporation of a menu system into the user interface in the virtual environment makes the interaction more versatile. A 3D menu

system is developed to provide different operations, such as changing the virtual object display mode, switching between different tasks, etc. The principles of the 3D menu system are directly related to those of the 2D menu system in the windows-based applications. The 3D menu system contains multiple entities, such as buttons, sliders, and dialog boxes, and entity hierarchy, such as pressing this button will pop up another menu. However, in the 2D menu user interface, only the mouse and keyboard can be used to interact with the menu. The 3D menu system is essentially a group of virtual objects floating in the virtual environment. It can be reached by the manipulator like a physical control panel, and can be invoked or closed by different hand gestures. Depending on the user's need, the 3D menu can stay fixed in front of the user's viewpoint, or fixed in the virtual world but the user can bring it to the view by using a simple gesture.

The *sensor* module contains the driver program to control the hardware devices. The hardware devices can be categorized as the tracking device, manipulator, and viewing device. The tracking device is used to track the user's head and hand positions such that when the user *walks around* the virtual world, the computer generated scenes will change correspondingly to simulate the user changing the viewpoint or moving his/her hand in the real world. The manipulator is a computer image that corresponds to real hand motion and is used to manipulate the virtual object directly to change the design shape or press a button on the virtual menu. The viewing device is used to display the computer generated scene to the user to create the immersive illusion of presence in the virtual world. The hardware peripherals used in this research are listed in Table 3.2. The application developed as a result of this research can use any combination of the following devices as VR interfaces.

The *graphic data base* module contains the vertex and polygon information for displaying the graphic objects. It is different from the information in the *FEA model data base* and the *analysis results data base* since it contains only the boundary mesh of the original Finland element model and the boundary contour information of the analysis results for the efficiency purpose of the graphic display. The *graphic visualization* module is used to display the scene of the virtual environment. It displays the graphic data as virtual objects, shows the deformation and color contour stresses, and updates the position and orientation of the viewpoint and the manipulator. The *user interface*, *sensor*, and *graphic visualization* modules are the only platform dependent modules in this program. In the future, if different computer platforms or different graphics software is needed, only these three modules need to be changed to accommodate the new system.

Table 3.2: Virtual reality hardware peripherals

DEVICE	DESCRIPTION	FUNCTION
Fakespace BOOM3C	display device with integrated motion tracking and navigation ability	viewing and navigation
n-Vision VGA Head Mounted Display equipped with an Ascension Flock of Birds tracker	display device with motion tracking	viewing
Ascension Flock of Birds	motion trackers	motion tracking
Fakespace Pinch Gloves	picking/navigation device	picking/navigation
Virtual Technologies Cyberglove	picking/navigation device	picking/navigation
StereoGraphics CrystalEyes	display device for stereo images	viewing

3.2.3 Free-Form Deformation

The *free-form deformation* module is used to find the deformed shape of the finite element model, determine the sensitivity coefficients of the shape design variables, and perform the interactive structural shape design. The internal modules, *bounding box definition*, *design area parametrization*, *design variable specification*, *what-if study*, *interactive design optimization* and *update design model*, combined with the *sensitivity analysis* global module which will be introduced in the next section, are the same design steps that were described in Section 3.1. The definition of the bounding box to construct the NURBS surface or volume is initially defined in an external configuration file. The definition includes the knot vectors of the NURBS and the initial positions of the control points. Other definitions of the virtual environment in the configuration file will be introduced in Appendix A. The functions in the *bounding box definition* module incorporate sensor information from the virtual environment sensor hardware to manipulate the bounding box and to specify the design area. After the bounding box is specified, the finite element nodes inside the bounding box will be parametrized through the inverse mapping method. The *design area parametrization* contains a numerical optimization solver to find the parametric value of the finite element nodes inside the bounding box. The *design variable specification* module contains the function to define the deformation method, depending on the choice of the design variable, and outputs the MSC/NASTRAN data file to compute the sensitivities. A sample data file generated by the design variable specification module is presented in Appendix B. In the *what-if study* module, the change of the design variable is controlled by the sensor. The sensor attached to the data glove indicates a change in position of the design variable. As the design variable changes, the stress and displacement of the new design is updated using linear Taylor series in the *update design*

model module. Since the update function of the linear Taylor series is easy and efficient to implement, the designer is able to obtain the approximated analysis results in real time without performing time-consuming finite element reanalysis. In the *interactive design optimization* module, the design variable can not be arbitrarily changed, but is subject to an optimal solution of a weighting function instead. The *update design model* is used to prepare the updated information of the virtual design model and the virtual analysis results model after the change of the design variable is determined. Details of this procedure are found in Chapter 6.

3.2.4. Sensitivity Analysis

Once the input data file for the sensitivity analysis is generated, the program will automatically create a subprocess to run the MSC/NASTRAN analysis. Since the sensitivity analysis may take considerable time, the processes for the rendering of the virtual environment and the MSC/NASTRAN analysis should be executed in parallel, otherwise, the simulation of the virtual design will stop as the analysis begins. In this program, a subprocess is created to perform the MSC/NASTRAN job. The MSC/NASTRAN data block DSCM2 contains the sensitivities of the structural responses, and the DESVCP data block contains the design velocity fields of both the boundary and interior finite element mesh. These two data blocks will be outputted to the ASCII file format based on the designer's request, and the ASCII files will be used in the *free-form deformation* module to perform the *what-if study* and the *interactive design optimization* tasks.

3.3 Conclusion and Future Development

The procedure to accomplish interactive structural design can be categorized into five design steps as outlined in Section 3.1. The overall program to construct the virtual environment contains over 10000 lines of codes and is organized as 4 global modules. The introduction of the functionality of each global modules and their local modules have been presented in this chapter.

The program developed in this project is the first prototype to apply virtual reality techniques to interactive structural shape design. It provides an easy and intuitive way to specify the design variables, perform the structural shape design, and view the corresponding analysis result changes in real-time. This program extends current VR applications in engineering design by combining analysis results, sensitivity information and fast design optimization techniques such that the designer can perform the analysis-based structural shape design inside the virtual environment. Limitations of the current version are listed as follows and should be addressed in future developments.

1. The virtual environment global module contains platform dependent local modules to control the hardware peripheral and the software program to construct the virtual environment. The virtual reality software, World Tool Kit (WTK), is currently used as the basis for this program. Ongoing research seeks to combine the WTK library with the CAVE library [20] such that this program can be executed in the C2 facility at Iowa State University in order to supply fully immersive interaction for several participants at once.

2. For the current version, all the finite element grids on the elements should be in global Cartesian coordinates, and mixing element types in the finite element model is not allowed.

More elements types and the capability to mix element types should be incorporated in the next version in order to provide a variety of choices of finite element mesh for more complicated shape design.

3. The objective of the virtual reality technique is to provide a simple, and natural interaction with the computer data. It is also the ultimate purpose of this project to perform structural shape design in an easy and intuitive way. The interaction capability in this program that a designer has right now is limited to the use of the data glove. More interaction techniques can be incorporated into this program to provide better manipulation capabilities in the virtual environment. For example, vocal command inputs can be used to define the precise movements of the design variable and sound feedback can provide intersection information when the data glove touches a virtual object.

CHAPTER 4. NURBS-BASED FREE FORM DEFORMATION

Besides the current applications of VR techniques, this research presents an innovative approach to the specific challenge of structural shape design. The goal was to develop VR software where the user can change the shape of a virtual object and examine the effect the shape change has on the structure deformation and stress distribution throughout the object. The approach integrates the analysis results, sensitivity information, and structural shape manipulation techniques such that a designer can easily relate shape changes to structural responses and quickly find a suitable shape that will meet deformation or stress constraints in an interactive, highly-immersive virtual environment. One important issue of such an application is to establish a mechanism to relate the structural shape design variable and the finite element model such that it not only provides an easy way to manipulate the structural shape but also provides information so that a linear stress and displacement approximation can be used to update the analysis results. In this chapter, the structural shape manipulation technique that will meet this goal is introduced. This technique, called *NURBS-based Free Form Deformation* (NFFD), provides an easy way to manipulate structural finite element models and also maintain the linear relationship of the finite element mesh changes with respect to the design variable in order to facilitate the linear approximation of the stress and displacement. In the following sections, the general background of NFFD and its direct manipulation technique are introduced. Some efficient algorithms for evaluating modified NFFD models are developed to facilitate the interaction in the virtual environment. The derived algorithms in this chapter are capable of dynamic real-time manipulation of NURBS-based curves, surfaces and solid mod-

els through modifying the control points or the geometric model directly. A unified approach is also presented for manipulating the NURBS model by simultaneously altering the weights and the coordinates of the control points. The performance of these algorithms is evaluated on a NURBS surface and NURBS solid model with various resolutions.

4.1 Background

The Non-Uniform Rational B-Spline, or NURBS, representation of geometric models is widely used in CAD/CAM and the computer graphics industry due to its flexibility and versatility [48]. Since Versprille [49] first extended B-splines to rational B-splines and cast the term *NURBS*, several attractive properties of NURBS have made them the primary parametric representation chosen for geometric modeling. NURBS offers a unified mathematical formulation for representing free-form shapes as well as analytical shapes. A NURBS curve, surface or solid can be manipulated to produce a variety of shapes by changing the control point coordinates, weights, or knot vectors that make up the NURBS. Most widely accepted geometry standards already provide NURBS as the geometric model representation for data exchange, such as IGES [50], PHIGS [51] and STEP [52], which further extends the popularity of NURBS.

In the field of surface modeling, a geometric model can be represented by several bivariate (u, v) tensor-product NURBS surface patches or trimmed-NURBS surface patches on its boundary, assuming internal homogeneity. In solid modeling, a geometric model can be parametrized by a trivariate (u, v, w) parametric such that not only the surface points but also the interior points of the model are uniquely specified. Sederberg and Parry [53] first proposed

the *Free-Form Deformation* (FFD) technique to manipulate the free-form shape with solid hyperpatches. In their work, the hyperpatch is defined as a trivariate tensor product Bernstein polynomial and the control point lattice is an orthogonal parallelepiped with uniform divisions. Coquillart [54] extended the FFD technique to *Extend Free-Form Deformation* (EFFD) such that the control point lattice is not restricted to parallelepiped shape, and additional continuity constraints across the adjacent EFFDs are maintained. Although the control point lattice can be arbitrarily shaped, the basic formulations of the deformation are essentially the same. It becomes natural for subsequent research to explore the FFDs with different blending functions to overcome the restriction imposed by using a Bézier volume, such as the excessive computations and numerical instability that accompany the higher order Bézier volumes. Griessmair and Pugathofer [55] used trivariate B-splines as blending functions, and Lasser [56] presented rational Bézier volumes for the FFD basis. Recently, Lamousin and Waggenspack [42] coupled FFD with NURBS basis functions in order to inherit beneficial NURBS properties for free-form manipulation. For example, the evaluation of these NURBS basis functions are computationally stable when the degree increases, the continuities across the hyperpatches can be automatically maintained, and the control point lattice can be unevenly divided.

Due to the complexity of the NURBS functions, the interactive modification of NURBS becomes a computationally intensive task. Most applications deal with this problem by performing necessary control point changes before calculating the updated geometry information [42]. Some applications may use lower resolution models for interactive form modification and finer resolution models for final display [57]. These methods do not offer real-time evaluation for modifying complex geometry. Several approaches have been developed to overcome

this difficulty in order to achieve dynamic real-time modification. Rogers and Satterfield [58] first presented an efficient algorithm for dynamically updating the B-spline surface in 1982. The algorithm precalculates the basis functions since the basis functions are independent of the geometric location of the control points. Rogers and Adlum [59] extended this algorithm for the NURBS surfaces in 1990 to limit the computation effort for interactive manipulation of the control point coordinates or weights. Au and Yuen [60] presented a unified approach to modify the control point coordinates and weights simultaneously, however, their method is limited to NURBS curve manipulation.

In the above methods, the user indirectly changes the surface shape by moving control points or modifying weight parameters. Direct manipulation, presented by Hsu, Hughes, and Kaufman [57], is another powerful tool for geometric shape deformation. This technique allows the designer to directly deform the object, resulting in more intuitive and more precise shape changes than modifying the control point lattice. A pseudoinverse is used to calculate the corresponding control point movement with respect to the movement of selected points on the object. General free-form deformation methods (FFD) are then used to update the remaining points. One disadvantage of this approach is that the pseudoinverse can be computationally intensive to evaluate. Therefore, to facilitate real-time interaction and improve the efficiency of the modification processes in the virtual environment, an efficient algorithm for direct manipulation is needed. This chapter presents the development of such an algorithm. Furthermore, the weighting factor will be incorporated into the direct manipulation method. The ability to change the weighting factors, which is not included in Hsu, Hughes, and Kaufman's work, serves to extend the variety of shape changes that can be formed interactively.

The algorithms presented in this chapter are extensions of Rogers and Adlum's incremental formulation [59] which will be discussed later. These new algorithms are more advanced and more versatile in the following ways:

1. Because of the general nature of the new algorithms, they can be applied to NURBS curves, surfaces, and volumes. They can also be used to determine the effect on the updated NURBS model in real time when the control points and weights are modified individually or simultaneously.
2. They can be used in both direct and indirect manipulation of the NURBS models.
3. The efficiency of the algorithms is improved by reducing the size of the basis matrix through *the local modification scheme* which will be discussed in section 4.2.4.
4. The algorithms extend the direct manipulation method to allow users to alter NURBS control point coordinates and weights in real time.

The following sections describe the general formulation of free-form NURBS and the direct manipulation technique. The improved algorithms will then be introduced and their performance will be discussed. The free-form NURBS volume is used as an example for the equation formulation and derivation. Note that the general equations derived in this chapter can be used in the real-time modification for NURBS curves and surfaces as well.

4.1.1 Mathematical Form of NFFDs

The Cartesian coordinates of a point S with a parametric value of (u, v, w) on a 3D NURBS volume with degrees of p, q, and r in three parametric directions respectively can be expressed as follows:

$$S(u, v, w) = \frac{\sum_{i=1}^{nu}\sum_{j=1}^{nv}\sum_{k=1}^{nw} N_{i,p}(u)N_{j,q}(v)N_{k,r}(w)W_{i,j,k}P_{i,j,k}}{\sum_{i=1}^{nu}\sum_{j=1}^{nv}\sum_{k=1}^{nw} N_{i,p}(u)N_{j,q}(v)N_{k,r}(w)W_{i,j,k}} \tag{4.1}$$

where *nu*, *nv*, and *nw* are the numbers of control points in the *u*, *v*, and *w* parametric directions, respectively. In order to facilitate the equation derivation in the later sections and for efficiency considerations, Equation (4.1) is simplified as a general equation of the form:

$$\boldsymbol{S} = \frac{(\boldsymbol{N} \otimes \boldsymbol{W})\boldsymbol{P}^T}{\boldsymbol{N}\boldsymbol{W}^T} \tag{4.2}$$

where $\boldsymbol{N}$ is the tensor product of the NURBS basis functions defined over the nonperiodic, nonuniform knot vectors in each parametric direction. Each component of vector $\boldsymbol{N}$, denoted as $N_{i,j,k}$, is a scalar product of the basis functions at the parametric location (u, v, w). Thus,

$$N_{i,j,k} = N_{i,p}(u)N_{j,q}(v)N_{k,r}(w) \tag{4.3}$$

$\boldsymbol{W}$ is a row vector of the control point weights, and $\boldsymbol{P}$ is a row vector of the control points. Each component of $\boldsymbol{P}$, $P_{i,j,k}$, and $\boldsymbol{W}$, $W_{i,j,k}$, are the Cartesian coordinate vector (x,y,z) and the associated weighting factor of the control point in the i^{th}, j^{th}, and k^{th} position of the three parametric directions, respectively. For example, a NURBS volume is constructed by

a control point lattice with *nu*, *nv*, and *nw* control points in the three parametric directions. Thus the vectors in Equations (4.2) are

$$\begin{aligned} \boldsymbol{N} &= [N_{1,1,1}, N_{1,1,2}, \ldots, N_{nu,nv,nw}] \\ &= [N_{1,p}N_{1,q}N_{1,r}, N_{1,p}N_{1,q}N_{2,r}, \ldots, N_{nu,p}N_{nv,q}N_{nw,r}] \\ \boldsymbol{W} &= [W_{1,1,1}, W_{1,1,2}, \ldots, W_{nu,nv,nw}] \\ \boldsymbol{P} &= [P_{1,1,1}, P_{1,1,2}, \ldots, P_{nu,nv,nw}] \end{aligned}$$

The order of the subindex i, j, k in the vectors $\boldsymbol{N}$, $\boldsymbol{W}$ and $\boldsymbol{P}$ should be consistent, and $i \in [1, nu]$, $j \in [1, nv]$, $k \in [1, nw]$.

Here we define a notation $\boldsymbol{A} \otimes \boldsymbol{B} = [A_x B_x]$, $x \in [1, n]$ as a scalar product of each component of two vectors $\boldsymbol{A}$ and $\boldsymbol{B}$ with the same dimension n. Therefore, $\boldsymbol{N} \otimes \boldsymbol{W} = [N_{1,1,1}W_{1,1,1}, N_{1,1,2}W_{1,1,2}, \ldots, N_{nu,nv,nw}W_{nu,nv,nw}]$.

Note that the general Equation (4.2) can represent a point on a curve, a surface, or a volume, and the only difference is the content of the vectors $\boldsymbol{N}$, $\boldsymbol{W}$ and $\boldsymbol{P}$. For example if Equation (4.2) is used to represent a NURBS surface,

$$\begin{aligned} \boldsymbol{N} &= [N_{1,1}, N_{1,2}, \ldots, N_{nu,nv}] \\ &= [N_{1,p}N_{1,q}N_{1,p}N_{2,q}, \ldots, N_{nu,p}N_{nv,q}] \\ \boldsymbol{W} &= [W_{1,1}, W_{1,2}, \ldots, W_{nu,nv}] \\ \boldsymbol{P} &= [P_{1,1}, P_{1,2}, \ldots, P_{nu,nv}] \end{aligned}$$

For general free-form deformation of NURBS curves, surfaces, or volumes, the designer first determines the knot vector(s) and the coordinates of the control points. The NURBS curves, surfaces or volumes are then constructed by mapping the entire parametric space into Cartesian space through the blending function of Equation (4.2). A variety of free-form

shapes can thus be obtained by manipulating the control points, weights, and knot vector(s). For the NFFD technique proposed by Lamousin and Waggenspack, the designer first constructs a control point lattice and the knot vectors of three parametric directions. The 3D object is embedded within the control point lattice by solving the inverse mapping problem to find the corresponding parametric value of each data point on the object. Detailed information of the inverse mapping technique will be discussed in 4.1.2. After all the parametric values of the data points are determined, the isoparametric mapping is used to evaluate the effect on the embedded object when the coordinates or weights of the control points are modified. The advantages of using NURBS as the basis function for the free-form deformation can be found in reference [42].

4.1.2 Inverse Mapping Technique

The inverse mapping technique refers to the algorithm that finds the parametric value of a point on the geometric model defined by NURBS. Due to the NURBS property that no general formulation is available for inverse point mapping, the determination of the parametric coordinates of a NURBS data point is often subject to numerical optimization search methods [48]. For a specified data point $\hat{S}$, the parametric value $(\hat{u}, \hat{v}, \hat{w})$ is determined by finding the optimal solution of the following form:

$$\begin{aligned} &\textit{Minimize} && \left\| S(\hat{u}, \hat{v}, \hat{w}) - \hat{S} \right\| \\ &\textit{Subject to} && 0 \le \hat{u}, \hat{v}, \hat{w} \le 1 \end{aligned}$$

where S is determined by Equation (4.2) and $\hat{S}$ is already known. The code used to solve the above optimization problem is CFSQP [61], which is C code for the feasible sequential qua-

dratic programming method. The initial searching point of $(\hat{u}, \hat{v}, \hat{w})$ is set to $(0.5, 0.5, 0.5)$. When the objective function of $\|S(\hat{u}, \hat{v}, \hat{w}) - \hat{S}\|$ is minimized, the design variables $\hat{u}$, $\hat{v}$, and $\hat{w}$ are the desired parametric coordinates.

4.1.3 Direct Manipulation Method

In design it is desirable and more intuitive to be able to directly manipulate the geometric model instead of moving the control points, thus the designer has better control over the exact deformed shape. The direct manipulation technique, proposed by Hsu, Hughes, and Kaufman [57], is presented to manipulate the free-form shape of the geometric model by modifying the model directly. The basic idea here is to calculate the displacements of the control points required to produce the prescribed displacements of the points on the model. From Equation (4.2), the displacement of a data point, $\Delta \boldsymbol{S}$, can be written as a function of the displacement of control point $\Delta \boldsymbol{P}$.

$$\Delta \boldsymbol{S} = \mathbf{B}(\Delta \boldsymbol{P})^T \tag{4.4}$$

where $\mathbf{B} = \dfrac{(\boldsymbol{N} \otimes \boldsymbol{W})}{\boldsymbol{N}\boldsymbol{W}^T}$ is the NURBS blending function. $\Delta \boldsymbol{P}$ is the unknown in this equation, and it is determined by

$$\Delta \boldsymbol{P} = (\mathbf{B}^{+} \Delta \boldsymbol{S})^T \tag{4.5}$$

where $\mathbf{B}^{+}$ is the pseudoinverse of $\mathbf{B}$. Detailed information on determining the pseudoinverse $\mathbf{B}^{+}$ can be found in reference [62]. Note that the direct manipulation method presented by Hsu,

Hughes, and Kaufman only determines the corresponding changes of the control point movements. The equations derived in this paper will be able to determine not only the control point movements but also the weight changes, thus the designer can take full advantage of the NURBS properties to obtain the free-form shape with more varieties.

4.2 Real-Time Modification of NFFDs

In this section, advanced algorithms for real-time modification of NFFDs are presented. These algorithms are similar to Rogers and Adlum's algorithms [59], but are extended to the free-form deformation of 3D NURBS volume and the free-form direct manipulation. In addition, the efficiency of the algorithms is also improved. Rogers and Adlum's algorithms use incremental equations to update the deformed object such that the equations do not recalculate any identical information during the modification process. The efficiency of the modification is then increased by precalculating this information and storing it in memory arrays for later use. In the incremental equations presented here, all the tensor products of the NURBS function, as seen in Equation (4.2), are in vector form. The components of the vector $\boldsymbol{N}$ contain all possible combinations of the basis functions in each parametric direction, and the components of the control point vector $\boldsymbol{P}$ and the weight vector $\boldsymbol{W}$ are arranged in the same order with respect to the corresponding component of $\boldsymbol{N}$. Extra memory spaces are needed to store the index number of the components in vectors $\boldsymbol{N}$, $\boldsymbol{P}$ and $\boldsymbol{W}$ mapping to the index number in each parametric direction. This step can be done in the beginning of the modification process. During the modification, the efficiency to evaluate the updated data points can thus be improved since the inner product of the two vectors in one loop is easier and more efficient to perform

than the tensor product in nested loops. The other advantage is that, if only one or a small number of control points or weights is modified, the size of the vectors N, P and W can be reduced for the effected components only, thus the efficiency of the inner product is improved. Rogers and Adlum deal with this problem by adding if-statements in the nested loops to avoid the calculation of the basis function multiplication. However, their method does not reduce the number of loop iterations. The algorithms presented in this paper reduce the number of loop iterations to update the data points by tabulating the index number of the selected control points and basis functions before the modification. Along with the local modification scheme which will be explained in the following section, the algorithms will further improve the efficiency during the modification process.

In the following sections, the incremental equations for real-time modification of NURBS volumes will be presented in detail. With minor changes in the basis function vectors, these equations can be readily applied for NURBS curve and surface modification.

4.2.1 Modifying Control Point Coordinates of NURBS Volumes

The incremental equation for NURBS surface modification presented by Rogers and Adlum can be readily extended for NURBS volumes by multiplying the effect of the basis function in the third parametric direction. From Equation (4.4), the updated data point S_{new} can be found by

$$S_{new} = S + \frac{(\hat{N} \otimes \hat{W})}{Sum} \Delta \hat{P}^T \tag{4.6}$$

where

$$Sum = NW^T = \sum_{i=1}^{nu} \sum_{j=1}^{nv} \sum_{k=1}^{nw} N_{i,p} N_{j,q} N_{k,r} W_{i,j,k} \tag{4.7}$$

which has already been calculated in the initial stage and has been stored in memory. $\hat{N}$ and $\hat{W}$ are the vectors of the NURBS basis function tensor product and the vectors of the weighting factors which corresponding to the modified control points, with dimensions of the number of modified control points. $\Delta\hat{P}$ is a row vector which contains the movement of the modified control point coordinates only. Note that if only one control point is modified, $\hat{N} \otimes \hat{W}$ is simply a scalar product. For example, if the control point $P_{i,j,k}$ is modified,

$$\hat{N} \otimes \hat{W} = N_{i,j,k} W_{i,j,k} = N_{i,p} N_{j,q} N_{k,r} W_{i,j,k}$$

After selecting the control points to be modified, users need only arrange a proper vector of $\hat{N} \otimes \hat{W}$, then the efficiency to update a new data point by Equation (4.6) results in an improvement over the straightforward method by using Equation (4.2).

4.2.2 Modifying Weighting Factors of NURBS Volumes

Again, based on Rogers and Adlum's algorithms, if only the homogeneous weight factors of the control points are changed for $\Delta\hat{W}$, and the coordinates remained the same, the updated data point becomes

$$S_{new} = \frac{Sum}{Sum_{new}} S + \frac{(\hat{N} \otimes \Delta\hat{W})}{Sum_{new}} \hat{P}^T \tag{4.8}$$

Since the weighting factors appear in both numerator and the denominator in Equation (4.2), both need to be updated when the weights are changed. Sum_{new} is fairly easy to compute as follows:

$$Sum_{new} = Sum + \hat{N}(\Delta\hat{W})^T \tag{4.9}$$

Sum is calculated using Equation (4.7) and users only need to provide the vectors of $\hat{N}$, $\hat{P}$ and $\Delta\hat{W}$ to calculate the updated data point.

4.2.3 Modifying Control Point Coordinates and Weighting Factors of NURBS Volumes

If the control points and their associated weights are changed simultaneously to $\hat{P}_{new}$ and $\hat{W}_{new}$, the incremental equation of the updated data point becomes

$$S_{new} = \frac{Sum}{Sum_{new}} S + \frac{\hat{N}(\hat{W}_{new} \otimes \hat{P}_{new} - \hat{W} \otimes \hat{P})^T}{Sum_{new}} \tag{4.10}$$

Equation (4.10) can be considered as the general equation for determining the updated geometric model by modifying the control point coordinates and the weighting factors. Note that the equation degenerates to Equations (4.8) and (4.6) if the control point coordinates and weights are fixed, respectively.

4.2.4 Local Modification Scheme

One advantage of the NURBS representation is its ability to handle localized deformation. For example, moving the control point $P_{i,j,k}$ only changes the data points on the object

in the interval of the parametric space $[u_i, u_{i+p+1})$, $[v_j, v_{j+q+1})$, and $[w_k, w_{k+r+1})$ due to the fact that the NURBS basis functions are zero if the parametric location is not within this interval. u, v, and w are the entries of the knot vectors in the three parametric directions. For multiple control point modification, it is easier to set up the valid interval as $[u_{imin}, u_{imax+p+1})$, $[v_{jmin}, v_{jmax+q+1})$, and $[w_{kmin}, w_{kmax+r+1})$, where *imin*, *jmin*, and *kmin* are the minimum index numbers, and *imax*, *jmax*, and *kmax* are the maximum index numbers of the modified control points. Thus the efficiency of the modification process can be improved such that the Equations (4.6), (4.8) and (4.10) only involve data points where the parametric values are located within the valid interval. This local modification scheme is especially important for real-time calculations to improve the efficiency of the direct manipulation of NFFDs. The following sections will discuss this method in more detail.

4.2.5 Direct Manipulation by Determining Updated Control Point Coordinates

The main purpose of the direct manipulation technique is to determine the updated control point coordinates with respect to the changes made to the selected data points on the geometric model, and determine the movement for the rest of the data points. The data point movements can be obtained in one step by substituting Equation (4.5) into Equation (4.6), i.e.,

$$\Delta S = \frac{(\hat{N} \otimes \hat{W})}{Sum}(\mathbf{B}^{+}\Delta\tilde{S}) \tag{4.11}$$

where $\Delta\tilde{S}$ is a set of prescribed data point displacements, and $\mathbf{B}^{+}\Delta\tilde{S}$ is essentially determining a least-squares solution of $\Delta\hat{P}$ [57], which implies that the overall movements of the control points are minimized. The index numbers i, j, and k of each component of the vectors

$\hat{N} \otimes \hat{W}$, $\hat{N}_{i,j,k}\hat{W}_{i,j,k}$, are dependent on the knot vector span index number [63] of the selected data points' parametric values. Let $uspan_{min}$, $vspan_{min}$, $wspan_{min}$ and $uspan_{max}$, $vspan_{max}$, $wspan_{max}$ be the minimum and maximum span index numbers of the parametric values of the selected data point set respectively, and $i \in [uspan_{min} - p, uspan_{max}]$, $j \in [vspan_{min} - q, vspan_{max}]$, and $k \in [wspan_{min} - r, wspan_{max}]$. Thus the sizes of the vectors $\hat{N}$ and $\hat{W}$ are reduced, and the efficiency of the incremental Equation (4.11) can be improved. Note that if only one data point movement is prescribed, **B** in Equation (4.4) reduces to a row vector, and the pseudoinverse can be easily found by

$$\mathbf{B}^{+} = \frac{\mathbf{B}^{T}}{\|\mathbf{B}\|^{2}} = \frac{Sum \cdot (\hat{N} \otimes \hat{W})^{T}}{\|\hat{N} \otimes \hat{W}\|^{2}} \tag{4.12}$$

Thus no more inverse processes are needed and the efficiency can be improved significantly. If multiple data point movements are prescribed, the solution process of $\mathbf{B}^{+}\Delta\tilde{S}$ can be found in Hsu, Hughes, and Kaufman [57].

4.2.6 Direct Manipulation by Determining Updated Control Point Coordinates and Weights

Since the direct manipulation method presented by Hsu, Hughes, and Kaufman determines the minimum changes of the control point coordinates, one may think that if the weight changes are also incorporated into the direct manipulation equation, the control point movements could be further reduced and more varieties of shape changes can be explored. In the following, the efficient direct manipulation algorithm will be presented to determine the minimum control point movements and weight changes.

From Equation (4.10), the incremental equation can be rearranged as

$$NW^T \Delta S = \hat{N}(\hat{W}_{new} \otimes \hat{P}_{new} - \hat{W} \otimes \hat{P})^T - \hat{N}(\hat{W}_{new} - \hat{W})^T S_{new}$$

or

$$NW^T \Delta S = \hat{N}(\Delta(\hat{W}\hat{P}))^T - \hat{N}(\Delta\hat{W})^T S_{new} \tag{4.13}$$

where $\Delta S = S_{new} - S$, $\Delta(\hat{W}\hat{P}) = \hat{W}_{new} \otimes \hat{P}_{new} - \hat{W} \otimes \hat{P}$, and $\Delta\hat{W} = \hat{W}_{new} - \hat{W}$. If more than one data point is prescribed, say the parametric values $u_1, u_2, \ldots, u_n$ of the selected data points $S_{new}(u_i)$, $i \in [1, n]$, are repositioned, and Equation (4.13) becomes a set of n equations as in matrix form:

$$\begin{bmatrix} N(u_1)W^T \Delta S(u_1) \\ N(u_2)W^T \Delta S(u_2) \\ \cdots \\ N(u_n)W^T \Delta S(u_n) \end{bmatrix} =$$

$$\begin{bmatrix} N_{1,1,1}(u_1) & N_{1,1,2}(u_1) & \cdots & N_{nu,nv,nw}(u_1) \\ N_{1,1,1}(u_2) & N_{1,1,2}(u_2) & \cdots & N_{nu,nv,nw}(u_2) \\ \cdots & \cdots & \cdots & \\ N_{1,1,1}(u_n) & N_{1,1,2}(u_n) & \cdots & N_{nu,nv,nw}(u_n) \end{bmatrix} \begin{bmatrix} \Delta(W_{1,1,1}P_{1,1,1}) \\ \Delta(W_{1,1,2}P_{1,1,2}) \\ \cdots \\ \Delta(W_{nu,nv,nw}P_{nu,nv,nw}) \end{bmatrix} -$$

$$\begin{bmatrix} N_{1,1,1}(u_1)S_{new}(u_1) & N_{1,1,2}(u_1)S_{new}(u_1) & \cdots & N_{nu,nv,nw}(u_1)S_{new}(u_1) \\ N_{1,1,1}(u_2)S_{new}(u_2) & N_{1,1,2}(u_2)S_{new}(u_2) & \cdots & N_{nu,nv,nw}(u_2)S_{new}(u_2) \\ \cdots & \cdots & \cdots & \\ N_{1,1,1}(u_n)S_{new}(u_n) & N_{1,1,2}(u_n)S_{new}(u_n) & \cdots & N_{nu,nv,nw}(u_n)S_{new}(u_n) \end{bmatrix} \begin{bmatrix} \Delta W_{1,1,1} \\ \Delta W_{1,1,2} \\ \cdots \\ \Delta W_{nu,nv,nw} \end{bmatrix}$$

or, expressed in matrix form:

$$[NW^T \Delta S]_{n\times 1} = [N]_{n\times np}[\Delta(WP)]_{np\times 1} - [NS_{new}]_{n\times np}[\Delta W]_{np\times 1} \qquad (4.14)$$

where np is the total number of control points, and np equals $nu \times nv \times nw$ for 3D NURBS volume cases. Note that the general equation above is essentially the same as the unified equation in reference [60], but the unified equation only works for NURBS curves. Since the multiple data points constraints have been prescribed, $[\Delta(\boldsymbol{WP})]$ and $[\Delta \boldsymbol{W}]$ are the only unknown column vectors in Equation (4.14) with dimensions of the total number of the control points, np. However, some components of these two vectors may become zero due to the local modification scheme which further reduces the number of the unknown variables.

The system of equations shown in Equation (4.14) can not be solved by using the pseudoinverse method as presented by Hsu, Hughes and Kaufman due to the nonlinear term $[\Delta(\boldsymbol{WP})]$ which appears on the right-hand side of Equation (4.14). Thus a numerical optimization search is performed to determine $\boldsymbol{W}_{new}$ and $\boldsymbol{P}_{new}$. A simple example using the Newton-Raphson method to find the optimal solution of Equation (4.14) is given in section 4.2.7.

4.2.7. Numeral Example for Direct Manipulation Method with Multiple Constraints

In this section, a simple numerical example is presented by using the Newton-Raphson method to solve the direct manipulation of a NFFD model with multiple data point constraints. Since the solutions for $\boldsymbol{W}_{new}$ and $\boldsymbol{P}_{new}$ in Equation (4.14) must be obtained using a numerical optimization search, it is not easy to perform real-time direct manipulation with several prescribed data points. Throughout this research, the single data point constraint

NFFD is performed to facilitate real-time interaction in the virtual environment, however, the solution process is given here for the more general case of multiple data point constraints which could be calculated if real-time interaction is relatively insignificant to the application.

A sphere is embedded inside a control point lattice as shown in Figure 4.1. The knot vectors that construct the 3D volume are $U = [0, 0, 1, 1]$, $V = [0, 0, 1, 1]$, and $W = [0, 0, 1, 1]$.

The control point coordinates are listed as follows:

$$P_{1,1,1} = \{0, 0, 0\} \qquad P_{1,1,2} = \{0, 0, 10\}$$
$$P_{1,2,1} = \{0, 10, 0\} \qquad P_{1,2,2} = \{0, 10, 10\}$$
$$P_{2,1,1} = \{10, 0, 0\} \qquad P_{2,1,2} = \{10, 0, 10\}$$
$$P_{2,2,1} = \{10, 10, 0\} \qquad P_{2,2,2} = \{10, 10, 10\}$$

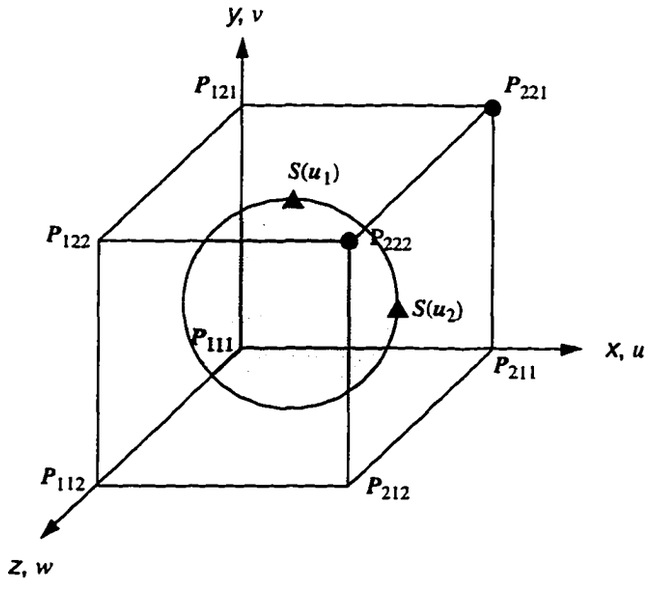


Figure 4.1: Numerical example of direct manipulation by determining control point coordinates and weights

and the initial weighting factors are all set to 1. Two data points on the sphere $S(u_1)$ and $S(u_2)$ are prescribed where the coordinates are $S(u_1) = \{5, 7.5, 5\}$ and $S(u_2) = \{7.5, 5, 5\}$ with the parametric values $u_1 = \{0.5, 0.75, 0.5\}$ and $u_2 = \{0.75, 0.5, 0.5\}$ originally. Here we want to determine the minimum changes of the control points $P_{2,2,1}$ and $P_{2,2,2}$ and their associated weights as the prescribed data points move to $S(u_1) = \{5, 8.13, 5\}$ and $S(u_2) = \{7.5, 6.03, 5\}$ for the direct manipulation. Note that these numbers are set up for the ease of the calculation in this example.

From the above statement, the only changes of the prescribed data points are in the y direction, thus we can solve the control point coordinate changes in the y direction only. Substituting the values into Equation (4.14), the system of equations becomes

$$\begin{bmatrix} 0.63 \\ 1.03 \end{bmatrix} = \begin{bmatrix} 0.0625 & 0.0625 & 0.1875 & 0.1875 & 0.0625 & 0.0625 & 0.1875 & 0.1875 \\ 0.0625 & 0.0625 & 0.0625 & 0.0625 & 0.1875 & 0.1875 & 0.1875 & 0.1875 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \Delta(W_{2,2,1}P_{2,2,1}) \\ \Delta(W_{2,2,2}P_{2,2,2}) \end{bmatrix}$$

$$- \begin{bmatrix} 0.5081 & 0.5081 & 1.5244 & 1.5244 & 0.5081 & 0.5081 & 1.5244 & 1.5244 \\ 0.3769 & 0.3769 & 0.3769 & 0.3769 & 1.1306 & 1.1306 & 1.1306 & 1.1306 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \Delta W_{2,2,1} \\ \Delta W_{2,2,2} \end{bmatrix}$$

Expanding the above matrix form results in

$$0.63 = 0.1875(W_{2,2,1}P_{2,2,1} - 10 + W_{2,2,2}P_{2,2,2} - 10) - 1.5244(W_{2,2,1} - 1 + W_{2,2,2} - 1)$$
$$1.03 = 0.1875(W_{2,2,1}P_{2,2,1} - 10 + W_{2,2,2}P_{2,2,2} - 10) - 1.1306(W_{2,2,1} - 1 + W_{2,2,2} - 1)$$

or

$$0.1875(W_{2,2,1}P_{2,2,1} + W_{2,2,2}P_{2,2,2}) - 1.5244(W_{2,2,1} + W_{2,2,2}) = 1.3312 \quad (4.15)$$
$$0.1875(W_{2,2,1}P_{2,2,1} + W_{2,2,2}P_{2,2,2}) - 1.1306(W_{2,2,1} + W_{2,2,2}) = 2.5188 \quad (4.16)$$

where $P_{2,2,1}$, $P_{2,2,2}$ are the new control point coordinates in the y direction, and $W_{2,2,1}$, $W_{2,2,2}$ are the new weighting factors. The process for solving the above nonlinear equations using the Newton-Raphson method is described below. The theoretical background of this method can be found in reference [64].

Set the residual functions for Equations (4.15) and (4.16) as

$$f_1 = 0.1875(W_{2,2,1}P_{2,2,1} + W_{2,2,2}P_{2,2,2}) - 1.5244(W_{2,2,1} + W_{2,2,2}) - 1.3312 \quad (4.17)$$
$$f_2 = 0.1875(W_{2,2,1}P_{2,2,1} + W_{2,2,2}P_{2,2,2}) - 1.1306(W_{2,2,1} + W_{2,2,2}) - 2.5188 \quad (4.18)$$

The goal is to find a value for $P_{2,2,1}$, $P_{2,2,2}$, $W_{2,2,1}$, and $W_{2,2,2}$ such that f_1 and f_2 will be reasonably small. This can be found by iteratively solving the below equation until f_1 and f_2 in Equation (4.19) are less than a small tolerance.

$$\begin{bmatrix} \frac{\partial f_1}{\partial P_{2,2,1}} & \frac{\partial f_1}{\partial P_{2,2,2}} & \frac{\partial f_1}{\partial W_{2,2,1}} & \frac{\partial f_1}{\partial W_{2,2,2}} \\ \frac{\partial f_2}{\partial P_{2,2,1}} & \frac{\partial f_2}{\partial P_{2,2,2}} & \frac{\partial f_2}{\partial W_{2,2,1}} & \frac{\partial f_2}{\partial W_{2,2,2}} \end{bmatrix} \begin{bmatrix} \Delta P_{2,2,1} \\ \Delta P_{2,2,2} \\ \Delta W_{2,2,1} \\ \Delta W_{2,2,2} \end{bmatrix} = \begin{bmatrix} -f_1 \\ -f_2 \end{bmatrix} \quad (4.19)$$

Set the initial value of $P_{2,2,1}$, $P_{2,2,2}$, $W_{2,2,1}$, and $W_{2,2,2}$ to 10, 10, 1, and 1 respectively. Substituting the values into Equation (4.19),

$$\begin{bmatrix} 0.1875 & 0.1875 & 0.3506 & 0.3506 \\ 0.1875 & 0.1875 & 0.7444 & 0.7444 \end{bmatrix} \begin{bmatrix} \Delta P_{2,2,1} \\ \Delta P_{2,2,2} \\ \Delta W_{2,2,1} \\ \Delta W_{2,2,2} \end{bmatrix} = \begin{bmatrix} 0.6300 \\ 1.0300 \end{bmatrix} \tag{4.20}$$

The above under-determined system of equations can be solved by the pseudoinverse method to find a least-squares solution [62]. We get $\Delta P_{2,2,1} = \Delta P_{2,2,2} = 0.7303$, and $\Delta W_{2,2,1} = \Delta W_{2,2,2} = 0.5078$. Thus $P_{2,2,1} = P_{2,2,2} = 10.7303$, and $W_{2,2,1} = W_{2,2,2} = 1.5078$. Substituting the new value of $P_{2,2,1}$, $P_{2,2,2}$, $W_{2,2,1}$, and $W_{2,2,2}$ into (4.17) and (4.18), we find $f_1 = 0.1390$ and $f_2 = 0.1389$ which are not close to zero. Thus the process is repeated again using the current values of $P_{2,2,1}$, $P_{2,2,2}$, $W_{2,2,1}$, and $W_{2,2,2}$ in Equation (4.19) until the convergence criteria is satisfied. The final answer is $P_{2,2,1} = P_{2,2,2} = 10.48$ and $W_{2,2,1} = W_{2,2,2} = 1.50$, achieved for a tolerance of 0.001.

4.2.8 Direct Manipulation of Single Data Point Constraint

The numerical optimization search often takes a considerable time to converge, thus it is not recommended for real-time manipulation processes. Here we propose a closed form solution for real-time manipulation of a single data point constraint in one coordinate direction. Assume the weight changes are proportional to their associated control point changes with a ratio of *r*. Thus,

$$\hat{W}_{new} = r(\hat{P}_{new} - \hat{P}) + \hat{W} \tag{4.21}$$

The above equation relates the variables $\hat{P}_{new}$ and $\hat{W}_{new}$ such that the unknown in this problem can be reduced to determine a proper updated control point vector $\hat{P}_{new}$ that satisfies the prescribed data point movements. The derivations of the equation to determine the vector $\hat{P}_{new}$ are shown below.

Taking Equation (4.10) and multiplying both sides by Sum_{new} results in

$$Sum_{new}\tilde{S}_{new} = Sum \cdot \tilde{S} + \hat{N}(\hat{W}_{new} \otimes \hat{P}_{new} - \hat{W} \otimes \hat{P})^T \tag{4.22}$$

Note that the data points $\tilde{S}$ and $\tilde{S}_{new}$ are scalars in the above equation since this development is only for the movement of one data point in one coordinate direction. Substituting Equation (4.9) into Equation (4.22) and rearranging the terms, we get

$$Sum(\tilde{S}_{new} - \tilde{S}) = \hat{N}(\hat{W}_{new} \otimes \hat{P}_{new} - \hat{W} \otimes \hat{P})^T - \hat{N}(\Delta\hat{W})^T\tilde{S}_{new}$$

or

$$\Delta\tilde{S} = \frac{\hat{N}}{Sum}(\hat{W}_{new} \otimes \hat{P}_{new} - \hat{W} \otimes \hat{P} - \Delta\hat{W}\tilde{S}_{new})^T \tag{4.23}$$

If we set **B** equal to $\frac{\hat{N}}{Sum}$ in Equation (4.23) and multiply both sides by the pseudoinverse $\mathbf{B}^+$, we get

$$\mathbf{B}^+\Delta\tilde{S} = \hat{W}_{new} \otimes \hat{P}_{new} - \hat{W} \otimes \hat{P} - \Delta\hat{W}\tilde{S}_{new} \tag{4.24}$$

Since this equation only works for manipulating one data point, the pseudoinverse $\mathbf{B}^{+}$ equals $\frac{Sum \cdot \hat{N}}{\|\hat{N}\|^2}$. Substituting Equation (4.21) into Equation (4.24), we have

$$(r(\hat{P}_{new} - \hat{P}) + \hat{W}) \otimes \hat{P}_{new} - \hat{W} \otimes \hat{P} - (r(\hat{P}_{new} - \hat{P}) + \hat{W})\tilde{S}_{new} + \hat{W}\tilde{S}_{new} = \mathbf{B}^{+}\Delta\tilde{S}$$

Rearrange these terms, we get the following equation

$$(r\hat{P}_{new} - r\hat{P} + \hat{W}) \otimes (\hat{P}_{new} - \hat{S}_{new}) - \hat{W} \otimes (\hat{P} - \hat{S}_{new}) = \mathbf{B}^{+}\Delta\tilde{S} \qquad (4.25)$$

where $\hat{S}_{new} = \{\tilde{S}_{new}\}$ is a row vector with a dimension equal to the dimension of $\hat{P}$ in order to maintain the dimensional consistency.

Expanding Equation (4.25) becomes

$$r\hat{P}_{new} \otimes \hat{P}_{new} - r\hat{P} \otimes \hat{P}_{new} + \hat{W} \otimes \hat{P}_{new} - r\hat{P}_{new} \otimes \hat{S}_{new} + r\hat{P} \otimes \hat{S}_{new} - \hat{W} \otimes \hat{P} = \mathbf{B}^{+}\Delta\tilde{S}$$

Rearranging the above equation, we get

$$r\hat{P}_{new} \otimes \hat{P}_{new} + (\hat{W} - r\hat{P} - r\hat{S}_{new}) \otimes \hat{P}_{new} + r\hat{P} \otimes \hat{S}_{new} - \hat{W} \otimes \hat{P} - \mathbf{B}^{+}\Delta\tilde{S} = 0 \qquad (4.26)$$

Based on the local modification scheme, we know the index number for each component of $\hat{P}$ and $\hat{W}$. Thus Equation (4.26) for each individual component in one coordinate direction can be written in scalar form:

$$r(\hat{P}_{new})^2 + (\hat{W} - r\hat{P} - r\tilde{S}_{new})\hat{P}_{new} + r\hat{P}\tilde{S}_{new} - \hat{W}\hat{P} - \mathbf{B}^{+}\Delta\tilde{S} = 0 \qquad (4.27)$$

Since the only unknown in Equation (4.27) is $\hat{P}_{new}$, solving the above quadratic equation, results in the following equation.

$$\hat{P}_{new} = \frac{-A \pm \sqrt{A^2 - 4rC}}{2r} \tag{4.28}$$

where $A = \pm\hat{W} - r\hat{P} - r\tilde{S}_{new}$, $C = r\hat{P}\tilde{S}_{new} \mp \hat{W}\hat{P} \mp B$. B in the expressions of A and C is the corresponding component of the vector $\mathbf{B}^{+}\Delta\tilde{S}$.

Note that the choice of the ± signs of Equation (4.28) will not only affect the value of the updated control point but also the updated weighting factors. Improper choice of the ± signs may cause unpredictable shape construction due to the improper application of the weighting factors. Normally, we would like the points on the geometric model to be pulled toward a particular control point as the change of that control point increases, thus the weight should also increase. Therefore,

$$\hat{P}_{new} = \frac{-A + \sqrt{A^2 - 4rC}}{2r} \tag{4.29}$$

and $A = \hat{W} - r\hat{P} - r\tilde{S}_{new}$, $C = r\hat{P}\tilde{S}_{new} - \hat{W}\hat{P} - B$.

If the movement of the data point is in the negative coordinate direction, the movements of the control points should also be in the negative direction. In order to maintain the positive increment of the weighting factor, the relationship of the updated weighting factors and the updated control point coordinates becomes

$$\hat{W}_{new} = r(\hat{P} - \hat{P}_{new}) + \hat{W} \tag{4.30}$$

Substituting Equation (4.30) into (4.25), the quadratic Equation (4.27) becomes

$$r(\hat{P}_{new})^2 + (-\hat{W} - r\hat{P} - r\tilde{S}_{new})\hat{P}_{new} + r\hat{P}\tilde{S}_{new} + \hat{W}\hat{P} + \mathbf{B}^{+}\Delta\tilde{S} = 0 \tag{4.31}$$

then the solution of the updated control point is

$$\hat{P}_{new} = \frac{-A - \sqrt{A^2 - 4rC}}{2r} \tag{4.32}$$

where $A = -\hat{W} - r\hat{P} - r\tilde{S}_{new}$, $C = r\hat{P}\tilde{S}_{new} + \hat{W}\hat{P} + B$.

Once the new control point set $\hat{P}_{new}$ is calculated by Equation (4.28), the control point movements can be readily obtained by $\Delta\hat{P} = \hat{P}_{new} - \hat{P}$, and the data points for the geometric model can be updated by

$$S_{new} = \frac{Sum}{Sum_{new}} S + \frac{\hat{N}(\ \mathbf{B}^{+}\Delta\tilde{S} + r\tilde{S}_{new}\Delta\hat{P}\ )^T}{Sum_{new}} \tag{4.33}$$

By using Equations (4.28) and (4.33), the updated model with minimum changes of control point coordinates and weights for one prescribed data point is determined. Note that r in these equations is serving as a user defined variable. Changing the value of r represents the relative ratio of weight changes to control point movements, and different shape changes with the same prescribed data point movement can thus be obtained. If $r = 0$, the update Equation

(4.33) is essentially the same as Equation (4.6), which means the weighting factors remain unchanged during the modification process.

Equation (4.33) assumes that the weight changes are directly proportional to the changes of the control point. Allowing the weights to be inversely proportional to the control point movements may cause numerical difficulties when a small change in the control point location occurs. Therefore, the inverse proportional relation is not used in this algorithm.

Note that Equations (4.28) and (4.33) are used for single data point constraint in one coordinate direction only. Generally, the data point movements are subject to change in multiple coordinate directions. In this case, Equations (4.28) and (4.33) can be used to calculate the weight change in the coordinate direction that has the most significant movement. Once the weighting factor change has been determined, Equation (4.10) can then be used to determine the control point movements and the model changes for the other coordinate directions.

Furthermore, the local modification scheme is applied again to update the rest of the data points on the object. The valid parametric value of the data point that will be affected by the selected data point is in the interval of $[u_{uspan_{min}-p}, u_{uspan_{max}+p+1})$,

$[v_{vspan_{min}-q}, v_{vspan_{max}+q+1})$, and $[w_{wspan_{min}-r}, w_{wspan_{max}+r+1})$.

4.3 Application and Discussion

In this section, the real-time NFFD algorithms derived previously are applied in a practical example to demonstrate the advantage in efficiency for geometry manipulation in a virtual environment. The software used to build the virtual environment is WorldToolKit (WTK) [44]. The hardware peripherals used in this example for interaction are the CyberGlove™

equipped with a Flock of Birds™ magnetic tracker, CrystalEyes™ stereo glasses, and a wall mounted projection system are used for the visual interface. The CPU time for updating the modified model by each equation is measured using a SGI Reality Engine R4400 machine with 150 Mhz processor. The CPU time is measured for the calculation of the spatial coordinates of the data points only, not including the rendering time for the graphic models and the interaction of the hardware peripherals.

4.3.1 Real-Time Free Form Deformation

A sports car model, shown in Figure 4.2 (a), is used to measure the CPU time for the NFFD process using the method described by Equation (4.2) without any precalculated information, the Rogers and Adlum's algorithms for modifying the control points and weights, and the methods presented in this paper. The sports car model is composed of one NURBS surface, with surface degrees of 3 in the u and v parametric directions and 34×44 control points as shown in Figure 4.2 (b). The free-form deformation of this NURBS surface becomes burdensome since the surface requires too many control point manipulations to make a smooth shape. Based on Lamousin and Waggenspack's method, the car model can be embedded into the control point lattice which contains fewer control points in order to simplify the manipulation process. Figure 4.2 (c) shows the control point lattice with only $7 \times 5 \times 3$ control points, and with degrees of 2 in the u, v, and w parametric directions. The control point $P_{4,3,3}$ is modified to compare the performance of these methods with different kinds of model resolution.

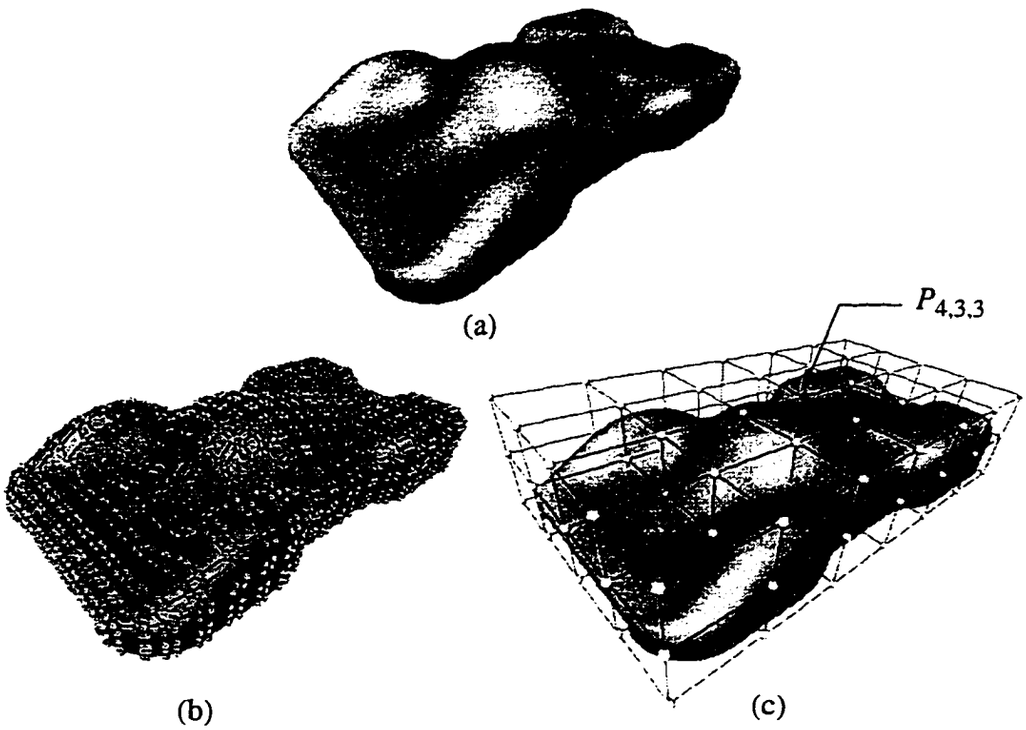


Figure 4.2: (a) Sports car model (b) Nurbs surface with control points
(c) NFFD with control point lattice

The comparison results are illustrated in Figure 4.3. The line indicated by circles is using Equation (4.2) to update all the data points on the geometric model without any precalculated information stored in the memory, thus the basis functions in Equation (4.2) are calculated repeatedly in every modification process. The computational effort of this algorithm depends on the number of data points, the number of control points, and the degree of the basis functions in each parametric direction. For the sports car model, the number of control points and the degree of the basis function remained the same, but the number of data points of the geometric model were changed to measure the CPU time for each method.

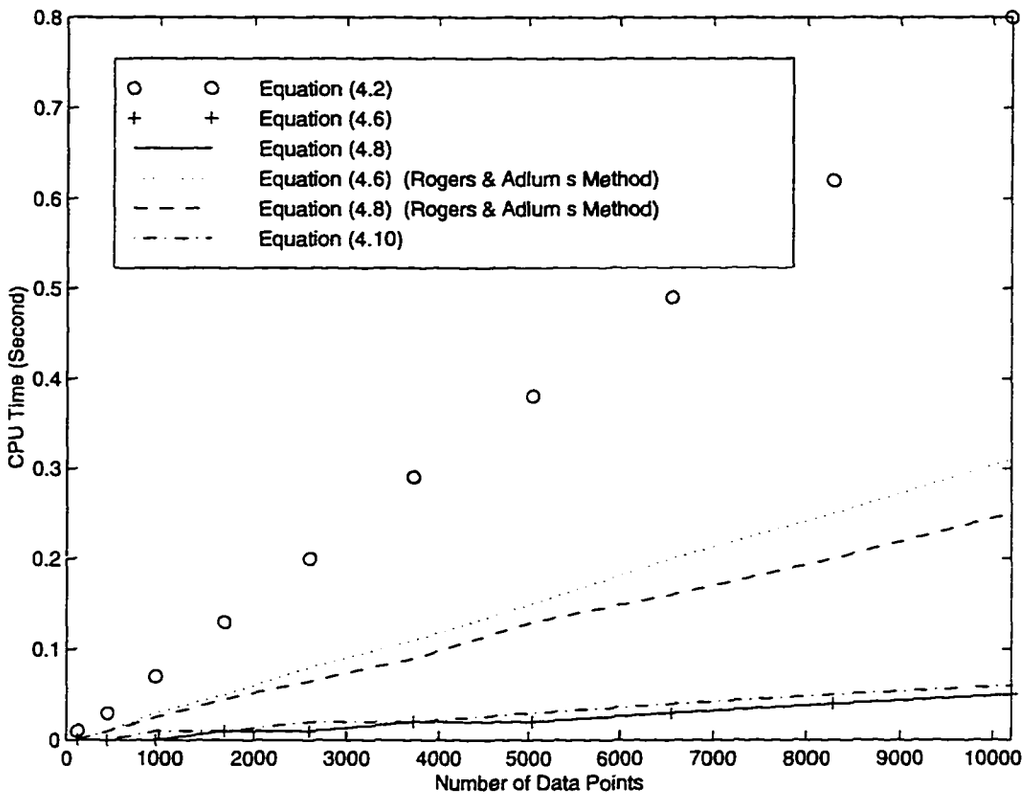


Figure 4.3: Comparison of the NFFD modification algorithms

The Rogers and Adlum's methods for Equation (4.6) and (4.8) in Figure 4.3 are the incremental algorithms presented by Rogers and Adlum to manipulate the control point coordinates (Equation (4.6)) and the weighting factors (Equation (4.8)), respectively. In Rogers and Adlum's algorithms, the if-statements are used in the nest for-loops to avoid unnecessary multiplication for the second term on the right hand side of the incremental Equations (4.6) and (4.8) to improve the efficiency of the modification process.

Figure 4.3 shows that the CPU time of all the methods is proportional to the number of data points. However, the efficient algorithms presented in this paper (Equations (4.6), (4.8) and (4.10)) have better computational efficiency due to the incremental equations which have

saved a lot of CPU time by avoiding basis function recalculation. Moreover, the algorithms here have better efficiency than Rogers and Adlum's algorithms since the vector product of the second term on the right hand side of Equations (4.6) and (4.8) becomes a scalar product due to a single control point modification. The ratio of the CPU time increment with respect to the increment of the data points is also improved due to the local modification scheme where only a small portion of the data point set is modified. Rogers and Adlum suggested that the efficiency of their dynamic algorithm is inversely proportional to the number of control points if the number of data points and the degree of the basis function remain constant. This property also holds for the dynamic algorithms presented here because the local modification scheme reduces the number of data points affected by a certain control point if the number of control points in this parametric direction increases.

Note that the CPU time plot shown in Figure 4.3 would be different depending on the number and location of the selected control points, however, for most cases, the algorithms presented here provide a more efficient NFFD model updating method than the straightforward methods used in most NFFD applications [42][57][60] and the incremental algorithms proposed by Rogers and Adlum [60].

4.3.2 Real-Time Direct Manipulation

The sports car model is again used to demonstrate the efficient direct manipulation algorithms. The finger tip of the CyberGlove™, as shown in Figure 4.4 (a), is used as an interface to deform the car model by pressing or picking a data point (see Figures 4.4 (b) and (c)).

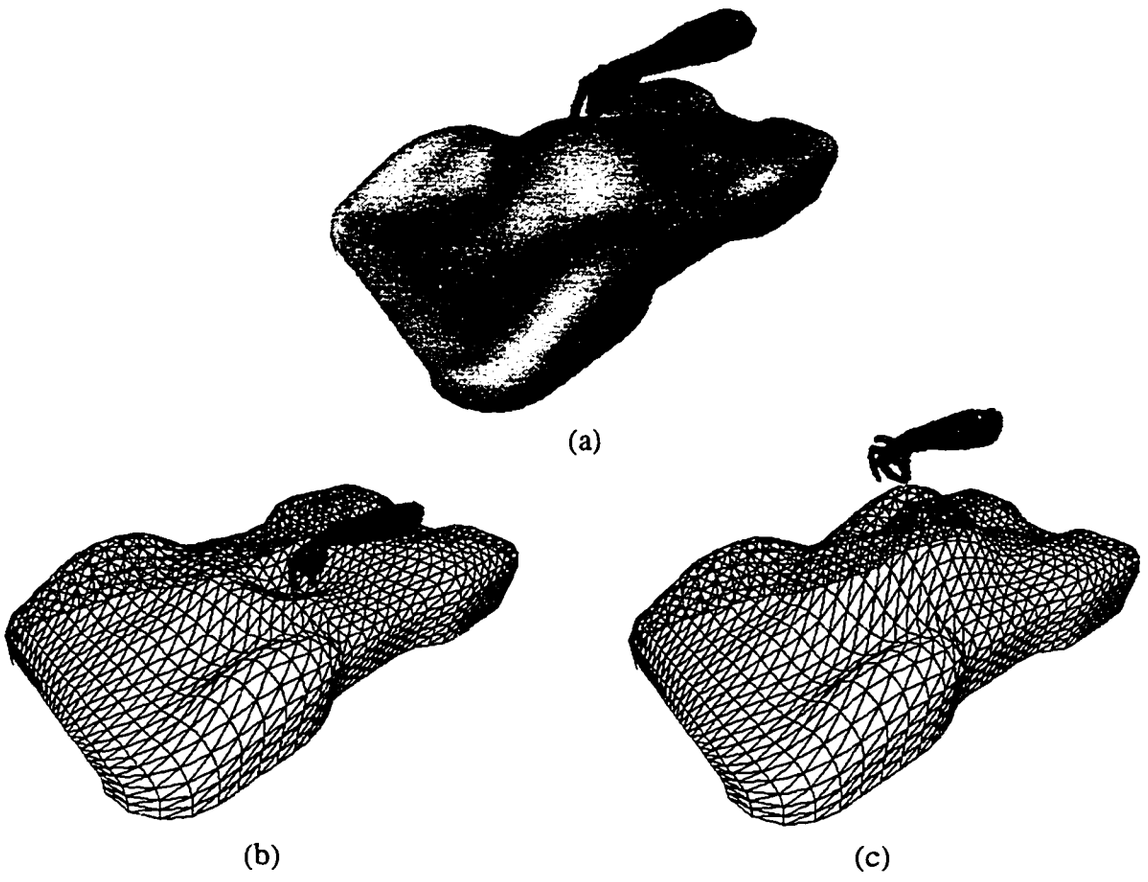


Figure 4.4: (a) CyberGlove and car model (b) Pressing manipulation
(c) Picking manipulation

By prescribing the same data point, the CPU times for the straightforward method and the method using Equation (4.11) are measured to compare the computational effort since both algorithms determine the minimum movement of the control points. The straightforward method described here is using Equation (4.5) to update the required control point movements, and Equation (4.2) to update all the data points without restoring any precalculated information.

Figure 4.5 shows that the method using Equation (4.11) saves almost 50% of the CPU time compared to the straightforward method, although it still depends on the selection of the data point.

For the efficient method of direct manipulation by determining updated control point coordinates and weights simultaneously, the variable r is the ratio of the weight changes with respect to the control point movement. It can also serve as the material elasticity of the geometric model. With different elasticity, the deformation of the model will be different. Although this elasticity is not exactly the same as the material property used in physics-based modeling, it provides a quick method to change the deformation of the geometric model for different properties by changing the design variable r. Figure 4.6 shows the deformations resulting from the same amount of data point movement with different values of r by using Equations (4.28) and (4.33).

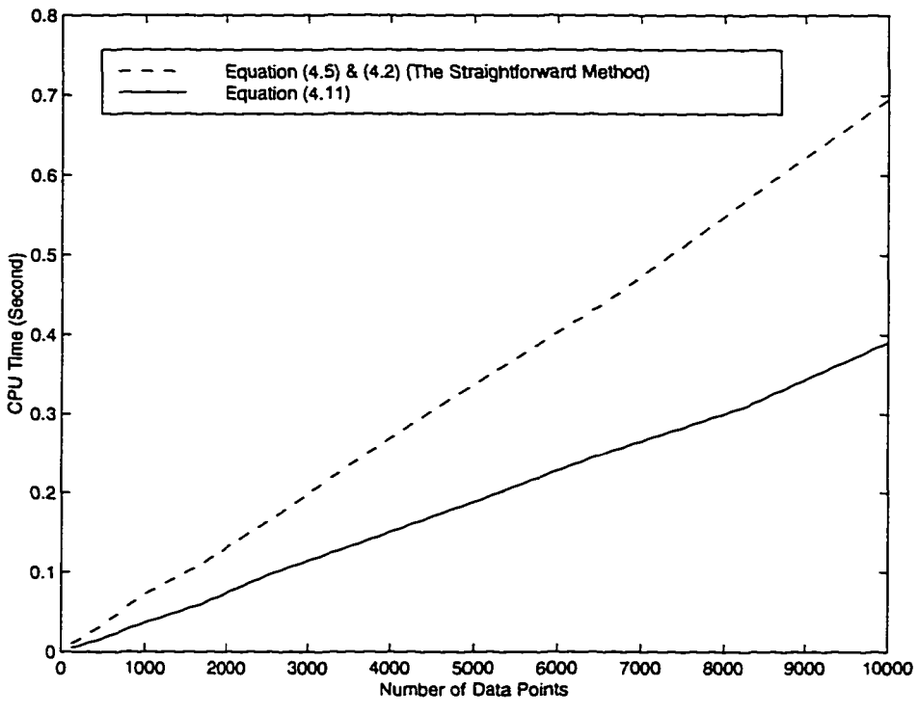


Figure 4.5: Comparison of the direct manipulation algorithms

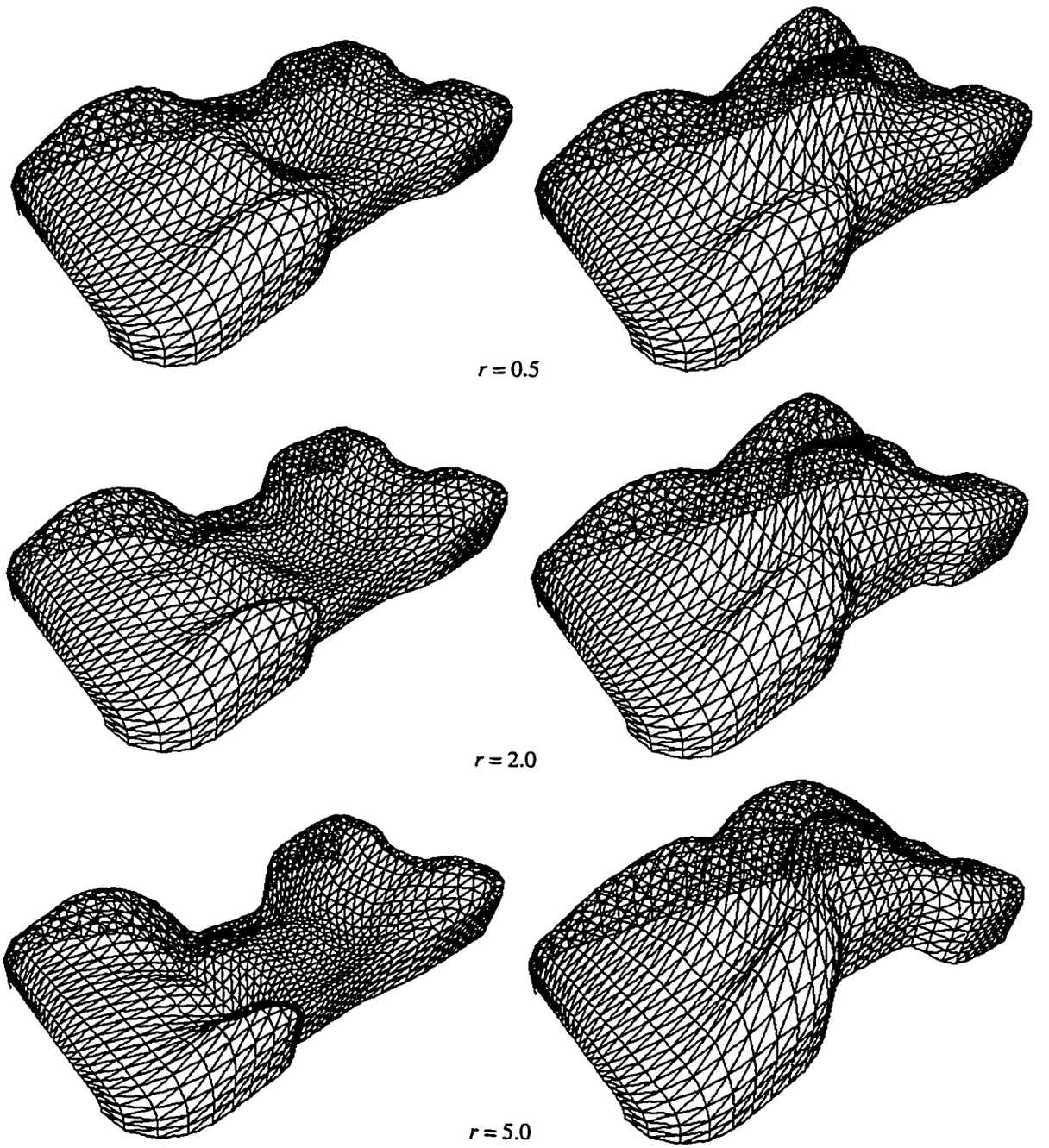


Figure 4.6: Car deformation with different values of r using Equations (4.28) and (4.33) for pressing and picking the center point (not shown) of the car model

4.4 Conclusion

Five efficient algorithms are presented for real-time NURBS-based free form deformation in this chapter. These algorithms are successfully implemented and compared to existing algorithms. The incremental equations of these efficient algorithms are suitable for the free-form deformation and direct manipulation of NURBS curves and surfaces, as well as 3D NURBS volumes. The algorithms are applied to a NFFD model to demonstrate the efficiency of the modification process since the free-form deformation of NURBS curves and surfaces are relatively easy. These algorithms exhibit sufficient performance for real-time modification by reducing the computational effort of the updated data points as much as possible. Thus the only thing that will delay the modification process is the graphic display of the geometric model.

Future work will focus on improving the efficiency of the direct manipulation method with multiple data points. The CyberGlove™ has been incorporated into the program as a picking device to manipulate the geometric model in a virtual environment. The interaction between the CyberGlove™ and the geometric model is still limited to the movement of one data point on the geometric model due to the limitation of Equations (4.28) and (4.33). The incremental equation should be further investigated for the approximated closed form solution of multiple data point constraints in order to provide a more efficient and more natural way of human hand and geometric object interaction.

CHAPTER 5. FINITE ELEMET BASED SHAPE DESIGN SENSITIVITY ANALYSIS

Finite element based shape design sensitivity analysis provides the derivatives of the structural responses, such as deformation, stress, and mode shape, with respect to the shape design variables of the finite element model. In design optimization, the sensitivity information is necessary to determine the searching direction in order to find the optimal shape using gradient-based optimization algorithms. It is also crucial information needed for examination of interactive trade-off design which uses sensitivity-based approximation techniques to avoid intensive reanalysis. An integrated virtual environment developed in this research can thus take the advantage of this approximation technique to facilitate the interactive structural shape design such that the designer can manipulate the shape of the finite element model and obtain the approximation of its structural responses in real-time. The general procedure used to calculate the shape design sensitivities of a finite element structural system can be summarized as (1) the choice of design variable, (2) the determination of the design velocity field, and (3) the computation of design sensitivities. This chapter presents a brief discussion of the most common methods for each step in the process, then follows with a discussion of the methods adopted in this research based on practical considerations of implementing this procedure in a virtual environment. The commercial finite element solution package MSC/NASTRAN [45] is incorporated into the interactive design process in VR in order to take the advantage of its capability in solving the domain velocity field and structural response sensitivities. The theoretical background of NASTRAN sensitivity analysis will also be introduced. A 2D cantilever

beam and a 3D engine connecting rod are used as examples to demonstrate the methods applied in the virtual environment.

5.1 Design Variable

5.1.1 Background

The choice of design variables is the initial and most important step in the process of the optimal shape design. Improper choice of design variables can lead to unpredictable and unacceptable results [65][66]. The shape design variable is used to describe the change of the structural shape as a function of a change in a design variable. The most common design variables employed in finite element based structural shape design include (1) the coordinates of the finite element nodes, (2) the coefficients of the polynomial or the basis shape vector to represent the boundary of the design model, (3) the control points of the spline curve or surface which represents the boundary of the design model, and (4) the dimension of generic primitives that define the design model.

5.1.1.1 Coordinates of the Finite Element Nodes

The use of the finite element nodal coordinates on the boundary of the design model provides a most simple and easy way to define the shape design variables in structural shape design optimization [67][68][69]. Movement of a node is used to define the change in the shape of the object. However, this method suffers from several drawbacks including mesh irregularities, the excessive number of design variables, and analysis inaccuracy [65]. Several researches have proposed alternative approaches to avoid such drawbacks. Cheu [70] used the coordinates of the specified master nodes as the design variables to reduce the number of

design variables of axisymmetric structural shapes for the shape design optimization process. Iman [71] proposed a *design element* concept to avoid the discontinuity of the boundary shape by grouping several finite elements as one design element, and to reduce the design variables by selecting a few master nodes to guide the change of the design element. In his approach, the design element is defined as a two or three dimensional isoparametric finite element using interpolation or blending functions to group several finite elements as one design element.

5.1.1.2 Coefficients of the Polynomial or the Shape Basis Vector

Polynomial representation is a natural choice for describing the continuous boundary of the design model with a simple mathematical form. The number of design variables can be reduced to the number of the polynomial coefficients [72][73]. Although this method provides a continuous representation of the boundary shape, oscillatory boundaries may occur if higher order polynomials are used to describe the shape [65]. Another approach defines the shape design variables as the coefficients of the linear combination of shape basis vectors [43][74]. However, proper shape basis vectors must be chosen in order to represent a variety of possible structural shapes.

5.1.1.3 Spline Representation

Spline representations eliminate the problem of oscillatory design boundaries, since they are composed of lower-order polynomial functions to maximize the smoothness of the design boundaries and extend the flexibility of the shape changes. It is similar to Iman's design element, however, the function to describe the boundary shape consists of the spline functions instead of finite element interpolation functions, and the design variable is not restricted to the finite element node but also the control points of the spline functions which allows changes to

the boundary shape in a global sense. The boundary continuity and regularity requirements are automatically taken into consideration and the mathematical form of the sensitivity derivatives can be easily established by the derivatives of the spline blending functions. Yang and Choi have shown that a better sensitivity accuracy can be achieved by using a spline representation rather than using a piecewise linear representation for the design boundary [75]. Various kinds of spline functions have been applied to represent the boundary shape. Luchi *et al.* [76] and Weck and Steinke [77] used cubic splines to define the boundary. Beaibant and Fleury [78], and Yao and Choi [79] applied Bézier and B-spline techniques to the finite element shape design. Schramm and Pikey [80] presented a geometry based approach by using the NURBS blending functions to represent the boundaries of the design elements.

5.1.1.4 Dimension of Generic Primitives

Another approach to structural shape design combines CAD technology with finite element solutions such that the dimensions of the geometric primitive, like width, length, and radius, can be defined as design variables. This method integrates the geometric design with analytical results to provide an easy way to guide the geometric shape changes with analysis considerations. Since additional information about the geometric definition of the design model needs to be provided, commercial CAD packages or finite element preprocessors are often necessary to provide such information by using generic primitives or some key dimensions to describe the geometry model. Some researches take advantage of automatic meshing capabilities to find the sensitivity information of the structural responses with respect to the design variable [81][82], however, the topology of the finite element mesh and the linear dependency of design variables are not easy to maintain. More recent approaches combine the

geometric model definition with the isoparametric mapping method [83][84] such that the regulation of the mesh topology can be maintained and the design velocity field can be easily found by the derivative of the mapping function. The isoparametric mapping approach is similar to Iman's design element concept, however, the difficulty in finding an appropriate mapping function can be alleviated by using a commercial geometric modeler which can provide a standard mapping function.

5.1.2 Choice of Shape Design Variable in the Interactive Virtual Environment

The choice of design variables adopted in this research derives from the spline representation and the design element concept such that the design variable can be not only the control point of the spline function but also any nodal point of the finite element model. The technique to relate the design variable with finite element shape changes is based on the use of NFFD.

The intent of the interactive structural shape design in the virtual environment is that it allows the designer to manipulate the free-form shape of any 2D or 3D finite element model as if it were clay in a sculptor's hands, and view the analysis results of the modified shape in real-time without performing reanalysis. As was discussed in Chapter 4, an innovative method is proposed to perform the structural shape changes by using NFFD technique. The NFFD method not only provides a versatile shape manipulation capability, but also offers a mathematical formula to determine the sensitivity of the mesh changes for the computation of the structural response sensitivities. It is natural to use the control points of the NFFD lattice or the finite element nodes of the embedded section as the structural shape design variable, since the designer can easily specify the design variable by picking the control points or the nodal points of the design model using a dataglove or any picking device in the virtual environment.

If the control point of the NFFD lattice is used as the design variable, the straightforward method using incremental Equation (4.6) is used to deform the shape of the embedded finite element nodal coordinates. If the finite element node in the embedded section is selected as the design variable, the direct manipulation technique using Equation (4.11) is used to change the structural shape. Note that the finite element node as the design variable is not restricted to some boundary master node as in Iman's design element concept, but can be any finite element node on the embedded section of the design model.

As was discussed in section 4.2.3 and 4.2.6, the weighting factor in NFFD provides an additional modification capability. Figure 5.1 shows an example where the weighting factors

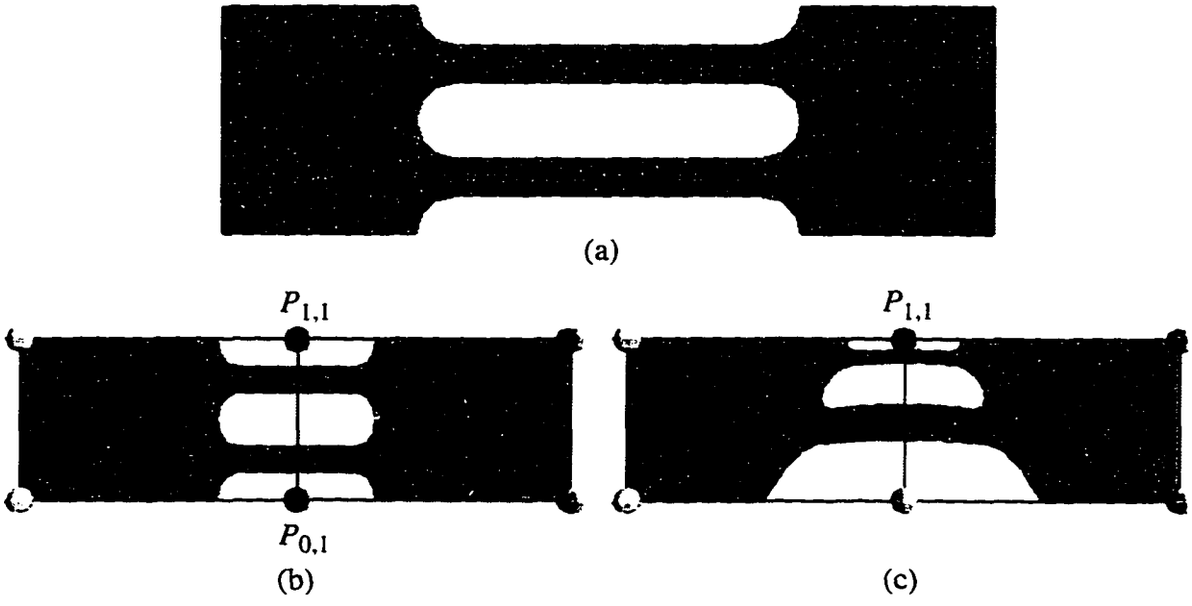


Figure 5.1: NFFD Method by modifying weighting factor(s)
(a) Finite element model of a beam with a slot
(b) Increasing weighting factors of $P_{0,1}$ and $P_{1,1}$ simultaneously
(c) Increasing weighting factors of $P_{1,1}$ only

are changed and the resulting effect on the shape of the finite element model is displayed. By changing the weighting factors of the control points $P_{0,1}$ and $P_{1,1}$ simultaneously, the size of the slot on a beam, as shown in Figure 5.1 (b), can be decreased as the weights increase. However, in general, changing the weighting factor may result in irregular finite element mesh or mesh distortion. Figure 5.1 (c) shows that the change of the weighting factor in $P_{1,1}$ results in an irregular mesh topology, and such a mesh will yield inaccurate analysis results. The improper application of the weights can also result in a bad parametrization and the subsequent shape changes through isoparametric mapping can be destroyed [48]. Thus the weighting factor will not be considered as the design variable in this research.

5.2 Design Velocity Field

5.2.1 Background

The design velocity field (DVF) refers to the movements of the material due to the change in the structural domain. From the finite element analysis point of view, it is the sensitivity of the finite element mesh changes with respect to the change of the shape design variable. DVF computation is important in determining shape design sensitivity coefficients and updating the finite element mesh for shape design optimization, since the accuracy of the structural response sensitivities and the optimal solution of the shape design is highly dependent on the results of DVF [85]. The computation methods for DVF can be categorized as (1) finite difference methods with automatic mesh generation technique, (2) isoparametric mapping methods, (3) boundary displacement and fictitious load methods, and (4) hybrid methods

of isoparametric mapping and boundary displacement techniques [86]. Detailed information on the theoretical and practical requirements of DVF can be found in reference [86].

5.2.1.1 Finite Difference Methods with Automatic Mesh Generation

Using finite different methods to determine the DVF of a structural system can be achieved by subtracting the nodal coordinates of the original finite element mesh from the updated mesh resulting from the perturbed design. The automatic mesh generator in most commercial finite element preprocessors provides an easy way to create the finite element mesh. However, the topology of the perturbed design is not easy to preserve, for example the number of the total finite element nodes for the perturbed model may not be the same as the original one, thus the finite difference method can not be properly applied to obtain the DVF. To overcome the inconsistency of the mesh topology, the Laplace smoothing technique can be used to find the updated mesh of a perturbed design [81][82]. Laplace smoothing is a recursive process to average the internal node coordinates and avoid mesh distortion after the initial mesh is generated by the mesh generator. The advantage of this method is that it is easy to implement and interface with a commercial CAD package. However, the linear dependency of the DVF with respect to the design variable is not easy to maintain since there is not a general mathematical formula to relate the design variable with mesh changes, thus the derived structural response sensitivities may not be reliable to predict the approximated results of the modified finite element model [86]. Furthermore, applying Laplace smoothing techniques for large finite element models is inefficient [87].

5.2.1.2 Isoparametric Mapping Method

The isoparametric mapping method is derived from the design element concept such that the finite element mesh is mapped into a fixed parametric space. As the design variables of the design elements change, the global Cartesian coordinates of the finite element model are updated using the same mapping functions from the corresponding parametric coordinates. Iman [71] and Botkin *et al.* [88][89] used the shape function of the finite element as the mapping function of the design element. Choi and Chang [90] used the cubic spline hyperpatch as the mapping function through the use of a commercial geometric modeler. Since the mapping function provides the explicit mathematical form of the geometry changes as the function of the design variables, it is efficient to obtain the DVF by finding the derivative of the mapping function, thus the linear dependency of DVF can be automatically maintained. One obstacle in the isoparametric mapping method is that a spatial decomposition of the design model is required for complicated geometry in order to divide the structure into relatively simple design elements. This process is tedious and inefficient since it is often done manually with the aid of a graphical interface to the finite element preprocessor.

5.2.1.3 Boundary Displacement and Fictitious Load Method

Boundary displacement and fictitious load methods determine the design velocity field by prescribing displacements [79][91] or prescribing loads [92][93]. In both specifications, the DVF is determined by solving the discretized finite element equilibrium equation of an auxiliary linear elasticity model. The regularity and linear dependency of the DVF are automatically satisfied [86], however, this method is inefficient since an additional finite element solution is required, and the flexibility of the boundary shape changes is limited.

5.2.1.4 Hybrid Method

To provide a more flexible and efficient way to specify the boundary shape and determine the DVF of a design model, Chang and Choi [90] proposed a hybrid method by coupling the isoparametric mapping method and the boundary displacement technique. The isoparametric mapping is used to find the boundary DVF since the boundary nodes of the finite element model can be represented as a mathematical form of bicubic polynomial hyperpatches through the use of commercial software. The boundary DVF is simply derived from the derivative of the algebraic coefficient matrix of the bicubic patches. The domain DVF of the interior finite element nodes is then determined by the boundary displacement method once the boundary displacement is specified. This method combines the advantages of both the isoparametric mapping method and the boundary displacement method. It is more general than the isoparametric mapping method applied for a complicated model shape for which a proper mapping function is not easy to find, and is more efficient than the boundary displacement method since the displacements of the boundary nodes are prescribed by the efficient isoparametric mapping method.

5.2.2 Determination of Design Velocity Field in the Interactive Virtual Environment

The isoparametric mapping method and the hybrid method are adopted in this research to determine the DVF in order to update the finite element mesh for the shape sensitivity analysis in an efficient and flexible manner. The designer can choose either of these two methods depending on the designer's judgement of the complexity of the design model. If the design model contains only simple geometry, the isoparametric mapping method can be used to determine the boundary and domain DVFs. Using the isoparametric mapping method to deter-

mine the DVF is an efficient approach since it does not require additional finite element calculation or automatic mesh generators but only results in several matrix multiplications.

There are some special cases that need to be accounted for however. Since the finite element nodes embedded inside the control point lattice are changed simultaneously, it is hard to maintain certain feature shapes inside the embedded section. For example a round hole in a beam may become an elliptic hole as the beam is deformed when using the isoparametric mapping method. When the designer wants to preserve certain features, such as a hole or a slot, on the design model, or constrain the movement of certain boundaries, he/she can use the spatial decomposition technique or local modification to avoid changing the feature on the design model. These limitations will be illustrated through examples later in this chapter.

This method does not always work for complicated shapes with multiple features, and then the hybrid method is chosen to determine the domain DVFs. The hybrid method uses an auxiliary finite element solution to solve for the domain velocity field. Although it is inefficient to determine the domain DVF using this method, it serves to allow the user to constrain certain finite element nodes and also preserve the mesh regularity and linear dependency with respect to the design variable. In the following section, the general equations to determine the DVF using the isoparametric mapping method are derived based on the NFFD definition. The equations of the hybrid method which uses a finite element solution to determine the domain DVF are also presented.

5.2.2.1 Isoparametric Mapping Method

The isoparametric mapping method for boundary and domain DVFs is used to determine the derivative of the mapping function that defines the embedded area. For the NFFD tech-

nique presented in this research, the mapping function is a NURBS blending function and the design variable can be either a control point of the NFFD or a nodal point of the finite element model. For the case where the control point is the design variable, the DVF can be determined by taking the derivative of Equation (4.1). Assume the design variable is $P_{x,y,z}$, then the design velocity field $V(S)$ becomes

$$V(S)_{3\times 1} = \left[\frac{dS}{dP}\right]_{3\times 1} = \left[\frac{\partial S(u,v,w)}{\partial P_{x,y,z}}\right]_{3\times 1} = \left[\frac{N_{x,p}(u)N_{y,q}(v)N_{z,r}(w)W_{x,y,z}}{Sum}\right]_{3\times 1} \tag{5.1}$$

where *Sum* in Equation (5.1) is the same as in Equation (4.7). Note that the derivative of the finite element nodal coordinates should be calculated with respect to the control point coordinates in the three Cartesian directions, thus the DVF of a specified finite element node S is a vector of three components.

For the case that a finite element nodal point $\hat{S}(\hat{u},\hat{v},\hat{w})$ is specified as the design variable, the design velocity field $V(S)$ can be found by differentiating the direct manipulation Equation (4.11).

$$V(S)_{3\times 1} = \left[\frac{dS}{d\hat{S}}\right]_{3\times 1} = \left[\frac{\Delta S(u,v,w)}{\Delta\hat{S}(\hat{u},\hat{v},\hat{w})}\right]_{3\times 1} = \left[\frac{(\hat{N}\otimes\hat{W})\mathbf{B}^{+}}{Sum}\right]_{3\times 1} \tag{5.2}$$

Since only one nodal point is specified as the design variable, it is similar to a single point constraint problem, thus the pseudoinverse $\mathbf{B}^{+}$ in Equation (4.2) can be calculated using Equation (4.12).

$$V(S)_{3\times 1} = \left[(\hat{N}\otimes\hat{W})\left(\frac{\hat{N}\otimes\hat{W}}{\|\hat{N}\otimes\hat{W}\|^{2}}\right)^{T}\right]_{3\times 1} \tag{5.3}$$

The DVFs in Equations (5.1) and (5.3) serve as entries for the DVGRID data block in the MSC/NASTRAN input file. The generated MSC/NASTRAN input file is used to calculate the structural response sensitivities in order to display the stress contour model of the modified design in real time. Equations (5.1) and (5.3) will also be used to update the finite element model mesh in the virtual environment through a linear Taylor series approximation.

For a complicated structural shape, the isoparametric mapping method is used to define the boundary DVF, and the boundary displacement method is used to specify the domain DVF. The combination of the isoparametric mapping method and the boundary displacement method is referred to as the hybrid method [86]. In this research, the boundary displacement method is solved using MSC/NASTRAN design sensitivity solution sequence. The theoretical background used in MSC/NASTRAN to determine the domain DVF is introduced below.

5.2.2.2 Boundary Displacement Method in the Hybrid Approach

The boundary displacement method computes an auxiliary linear elasticity problem with prescribed boundary displacements to determine the domain DVF. Without additional external forces but a given displacement at the boundary, the same finite element code used for analysis can be used to solve for the domain DVF. Due to the ease of incorporating the boundary displacement method into the existing finite element code, most commercial finite element software packages use this method to solve for the DVF during the shape design optimization process. The discretized equilibrium equation of a finite element model can be written as

$$[K]\{V\} = \{f\} \tag{5.4}$$

where $[K]$ is the reduced stiffness matrix of the auxiliary structure, $\{V\}$ is a vector of the DVF, and $\{f\}$ is the unknown boundary force which produces the perturbation of the prescribed boundary DVF. Equation (5.4) can be rewritten in the partitioned form as

$$\begin{bmatrix} K_{bb} & K_{bd} \\ K_{db} & K_{dd} \end{bmatrix} \begin{bmatrix} V_b \\ V_d \end{bmatrix} = \begin{bmatrix} f_b \\ 0 \end{bmatrix} \tag{5.5}$$

where $\{V_b\}$ is the boundary DVF, $\{V_d\}$ is the domain DVF, and $\{f_b\}$ is the fictitious boundary force applied on the varying boundary. Rearranging Equation (5.5), we get

$$[K_{dd}]\{V_d\} = -[K_{db}]\{V_b\} \tag{5.6}$$

Equation (5.6) defines a linear relationship between the boundary DVF and domain DVF. Since the boundary DVF can be specified by the isoparametric mapping technique, the domain DVF can be readily determined by the above equation. In MSC/NASTRAN, both boundary and domain DVFs can be output into a data block DESVCP. This data block can then be used to update both the boundary and interior mesh through a linear Taylor series.

5.3 Design Sensitivity

5.3.1 Background

Design sensitivity refers to the derivatives of the structural responses with respect to the design variables. Most gradient-based optimization algorithms and trade-off analysis require design sensitivities in searching for optimal designs to update the analysis through approxima-

tion techniques. Several methods have been developed to compute the sensitivity information for structural responses. The most popular methods, including (1) finite difference method, (2) analytical method (3) semi-analytical method, and (4) variational method, will be discussed in the following.

5.3.1.1 Finite Difference Method

Finite difference method has the advantage of being simple in concept, and easy to implement to obtain design sensitivities [81][88]. An additional finite element solution is used to solve for the perturbed structure due to a small change in the design variable. The difference of the analysis results between the original and the perturbed structures divided by the change of the design variable provides the linear sensitivity of the structural responses. Since this method needs little knowledge of the interior structure of the finite element model or the solution sequence, it is straightforward to find this sensitivity information using any commercial finite element package. This method, however, is inefficient due to the requirement of an additional finite element solution. In addition, the accuracy of the sensitivity calculations often is dependent on the choice of the perturbation step size.

5.3.1.2 Analytical Method

The analytical method is to differentiate the finite element system of equations directly with respect to the design variable [94][95]. It is straightforward in terms of mathematical derivation if the explicit expression of the mass matrix or the stiffness matrix can be obtained. However, for most structures, the mathematical form of the differential system of equations may result in a lengthy derivation and a highly complicated form. Thus this method is often

used for simple structural systems or typically for finite element program developers. The mathematical formulations of the analytical methods can be found in references [94][96].

5.3.1.3 Semi-Analytical Method

The semi-analytical method is proposed in order to improve the efficiency of the finite difference method and reduce the difficulty of the analytical method. It employs the finite difference method to obtain the derivative of the stiffness matrix or mass matrix, since the construction process of these matrices for the original and perturbed structures are identical but have different numerical results. The efficiency of the finite difference process can be significantly improved since the reconstruction of system equations for the perturbed structure can be eliminated. The analytical method is then used to obtain the sensitivity of the structural responses by solving a simple matrix equation. This is one of the most attractive methods in practical problems because of its generality and easy implementation [97] to be integrated in an existing finite element solution. Detail information of this method will be discussed in section 5.3.2.

5.3.1.4 Variational Method

Instead of differentiating the discretized finite element equation for the analytical and semi-analytical methods, the variational method differentiates the governing equations before discretization. The relationship between a shape variation of a continuous domain and the resulting variation of the structural responses can be described by the material derivative of continuum mechanics [94][98]. There have been studies showing that the variational methods are more accurate and efficient [98][99][100] then the discrete analytical methods. The only

disadvantage is that these methods require additional programming since they are not generally commercially available.

5.3.2. Computation of the Response Sensitivity in the Interactive Virtual Environment

In MSC/NASTRAN, the semi-analytical method is available in the design sensitivity analysis solution sequence to compute the structural response sensitivities. For the static finite element problem, the structural response sensitivity can be computed by differentiating the equation of static equilibrium of

$$[K]\{U\} = \{F\} \tag{5.7}$$

where $[K]$ is the stiffness matrix representing the structure, $\{U\}$ is the displacement vector of the finite element nodal coordinates, and $\{F\}$ is the vector of the external force. The differential equation with respect to the design variable x_i becomes

$$[K]\frac{\partial\{U\}}{\partial x_i} = \frac{\partial\{F\}}{\partial x_i} - \frac{\partial[K]}{\partial x_i}\{U\} \tag{5.8}$$

The right-hand side of Equation (5.8), which is often referred to as the pseudo-load vector, includes the partial derivatives of the vector $\partial\{F\}/\partial x_i$ and the matrix $\partial[K]/\partial x_i$ with respect to the design variable x_i. Since the vector $\{F\}$ and the matrix $[K]$ are generally implicit functions of the design variable, it is easier to use the finite difference method to approximate the derivatives. The vector $\{F\}$ and the matrix $[K]$ of the perturbed design model can be obtained using the same solution sequence without additional construction of the vector and matrix, thus the efficiency of the sensitivity computation is improved. After the pseudo-load is

determined, the displacement sensitivity is computed using Equation (5.8). Since the structural response $\{R\}$ is related to the design variable x_i and the displacement vector $\{U\}$, the derivative of $\{R\}$ can be approximated by using finite differences as

$$\frac{\partial\{R\}}{\partial x_i} \cong \frac{\{\Delta R\}}{\Delta x_i} = \frac{\{R(x_i + \Delta x_i,\, U + \Delta U) - R(x_i,\, U)\}}{\Delta x_i} \tag{5.9}$$

where ΔU in equation (5.9) can be obtained by

$$\{\Delta U\} = \frac{\partial\{U\}}{\partial x_i} \times \Delta x_i \tag{5.10}$$

The above equations briefly describe the computational theory behind the MSC/NASTRAN solution sequence `SOL 200` for the sensitivity calculation of the static response. Additional information, such as the computational methods of the eigenvalue sensitivities or buckling sensitivity, can be found in references [45][101].

The sensitivities obtained from MSC/NASTRAN are then used in a Taylor series to approximate the structural responses of stresses and displacements at each finite element node:

$$\sigma(x + \Delta x) = \sigma(x) + S_\sigma \cdot \Delta x$$

$$D(x + \Delta x) = D(x) + S_D \cdot \Delta x$$

where

$\sigma(x + \Delta x)$, $D(x + \Delta x)$ = approximate stress, σ, and displacement, D, at the design variable $x + \Delta x$

$\sigma(x)$, $D(x)$ = stress and displacement at the original design variable x obtained from finite element solution

S_σ, S_D = stress and displacement sensitivities

Δx = change in the design variable

5.4 NFFD and Shape Design Sensitivity Analysis

The NFFD presented in Chapter 4 has been used as the deformation technique for geometric models. In this research, the NFFD technique is used to deform the finite element model and obtain the sensitivities of the structural responses. Due to the linearity and regularity considerations, the finite element model can not be arbitrarily deformed as a free from shape of a geometric model. Some limitations of the NFFD are required to obtain a meaningful shape. The NFFD techniques for the finite element model shape design are introduced in this section. A 2D cantilever beam with a round hole is used as an example to demonstrate the NFFD process of the finite element model shape design in the virtual environment. A variety of shape changes can be explored by simply changing the control points or the finite element nodes. The continuity across the boundary of the design area can be easily maintained by a proper choice of the knot vectors. In this example, various deformation methods for the finite element model will be presented including global and local deformation, direct and indirect manipulation, feature preserving techniques, and the reduced basis method.

5.4.1 Global and Local Modifications

To perform the NFFD shape design, the designer first defines the parameters of the NURBS surface or volume which constructs the bounding box, such as the knot vectors and the location of the control points. The designer can then specify the design area and embed the desired section inside the NURBS surface or volume for the free form deformation. The glo-

bal and local modifications of the finite element model depend on the bounding box that embeds the design area. Figure 5.2 illustrates the process to specify the design area through the use of a PinchGlove, and a virtual menu system is provided to select user actions. Figure 5.2 (a) shows that, by default, all the finite element model is contained inside the bounding box for global modification. The designer can use either the PinchGlove to resize the bounding box directly (Figure 5.2 (b)), or the menu button for precision changes of the bounding box for local modification (Figure 5.2 (c)). Once the bounding box is specified and the

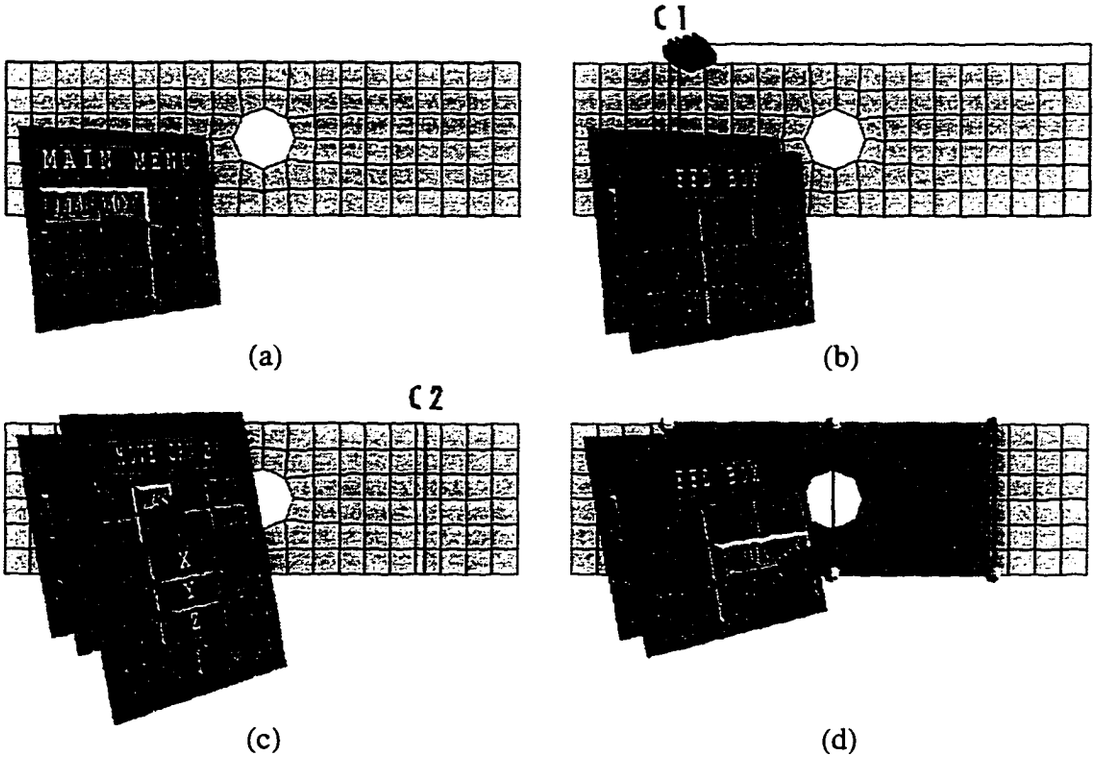


Figure 5.2: Process to specify the design area
(a) Press **FFD BOX** button to bring up the bounding box
(b) Use PinchGlove to resize the bounding box directly
(c) Use menu button to resize the bounding box
(b) Press **OK** button to perform parametrization

designer presses the button **OK**, the design area will be highlighted signalling that the parametrization between the finite element model and the NURBS surface or volume is complete (Figure 5.2 (d)).

The designer then selects a control point or a finite element node as the design variable. Figure 5.3 shows the shape change for both global and local modification of a 2D cantilever beam with a simple knot vector definition of $\mathbf{U} = \{0.0, 0.0, 0.0, 1.0, 1.0, 1.0\}$ and $\mathbf{V} = \{0.0, 0.0, 1.0, 1.0\}$. The degrees in parametric directions U and V are 2 and 1 respectively. Note that since the control points $P_{1,1}$ moves in the y direction (or V parametric direction) only, it is best to keep the degree of 1 in this direction in order to maintain the linear dependency of the finite element mesh movements for the linear sensitivity analysis.

For local modification, preserving continuity across the design boundary is sometimes desired. Using NFFD, this can be done by changing the knot vectors of **U** and moving the control point which will not affect the derivative of the boundary. The following derivation will present the case where the derivative of $\partial S(u, v)/\partial u = 0$ at $u = 0$ and 1, i.e., the slope at the boundary of the NURBS surface or volume is zero in order to maintain the continuity.

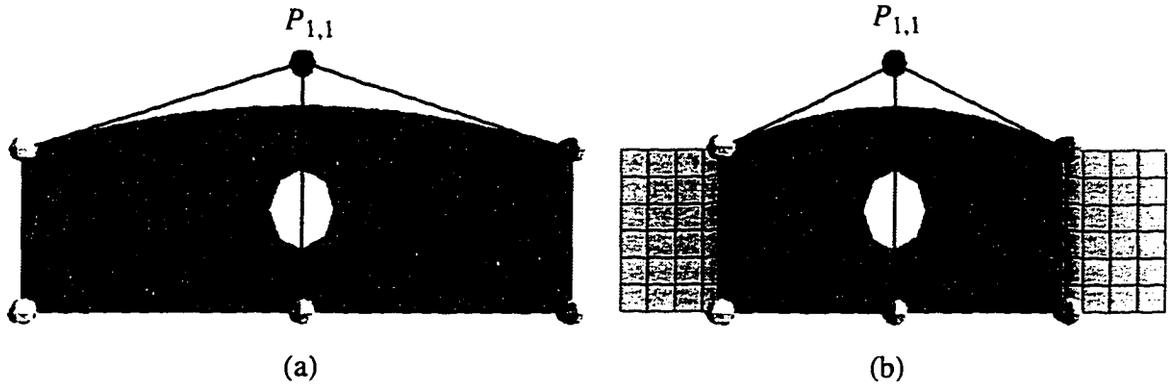


Figure 5.3: Shape changes for (a) Global modification (b) Local modification

The coordinate S of the finite element nodes embedded inside the design area can be defined by a NURBS surface function of

$$S(u, v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)W_{i,j}P_{i,j}}{Sum(u, v)} \tag{5.11}$$

where

$$Sum(u, v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)W_{i,j} \tag{5.12}$$

Take the derivative of Equation (5.11) with respect to u,

$$\frac{\partial S(u, v)}{\partial u} = \frac{Sum(u, v)\frac{\partial}{\partial u}A(u, v) - \left(\frac{\partial}{\partial u}Sum(u, v)\right)A(u, v)}{(Sum(u, v))^2} \tag{5.13}$$

where $A(u, v)$ is the numerator of Equation (5.11), and $A(u, v) = Sum(u, v) \cdot S(u, v)$.

Equation (5.13) becomes

$$\frac{\partial S(u, v)}{\partial u} = \frac{\frac{\partial}{\partial u}A(u, v) - \left(\frac{\partial}{\partial u}Sum(u, v)\right)S(u, v)}{Sum(u, v)} \tag{5.14}$$

From equations (3.4) and (3.20) in reference [63], we get

$$\frac{\partial}{\partial u}A(u, v) = \sum_{j=0}^{m} N_{j,q}(v)\left(\frac{\partial}{\partial u}\sum_{i=0}^{n} N_{i,p}W_{i,j}P_{i,j}\right)$$

$$= \sum_{i=0}^{n-1}\sum_{j=0}^{m} N_{i,p-1}(u)N_{j,q}(v)\left(p\frac{W_{i+1,j}P_{i+1,j} - W_{i,j}P_{i,j}}{u_{i+p+1} - u_{i+1}}\right) \tag{5.15}$$

and

$$\frac{\partial}{\partial u}Sum(u, v) = \sum_{j=0}^{m} N_{j,q}(v)\left(\frac{\partial}{\partial u}\sum_{i=0}^{n} N_{i,p}W_{i,j}\right)$$

$$= \sum_{i=0}^{n-1}\sum_{j=0}^{m} N_{i,p-1}(u)N_{j,q}(v)\left(p\frac{W_{i+1,j}-W_{i,j}}{u_{i+p+1}-u_{i+1}}\right) \qquad (5.16)$$

In order to make Equation (5.14) zero at $u = 0$ and $u = 1$ for $v \in [0, 1]$, Equations (5.15) and (5.16) should be zero at $u = 0$ and $u = 1$. Thus for any nonzero $N_{i,p-1}(0)$ and $N_{i,p-1}(1)$, $W_{i+1,j}P_{i+1,j} - W_{i,j}P_{i,j}$ and $(W_{i+1,j} - W_{i,j})$ should be zero in Equations (5.15) and (5.16) respectively, thus $P_{i+1,j} = P_{i,j}$ and $W_{i+1,j} = W_{i,j}$. For example,

$$\mathbf{U} = \{0.0, 0.0, 0.0, 0.0, 0.5, 1.0, 1.0, 1.0, 1.0\}$$

or

$$\mathbf{U} = \{0.0, 0.0, 0.0, 0.333, 0.666, 1.0, 1.0, 1.0\}$$

$N_{0,p-1}(0) = 1$ and $N_{3,p-1}(1) = 1$ for the above knot vectors, and the rest of the basis functions are zero. As shown in Figure 5.4, the y coordinates of the control points $P_{0,1}$, $P_{1,1}$, $P_{3,1}$ and $P_{4,1}$ are kept the same with the same weighting factor of 1. The designer can move the control point $P_{2,1}$ freely and maintain the continuity across the design boundary.

5.4.2 Direct and Indirect Manipulations

The direct and indirect manipulations depend on the selected design variable for the NFFD. As was discussed in section 5.1.2, for indirect manipulation, the control point of the NFFD is defined as the design variable while for direct manipulation, the finite element node serves as the design variable. The update method for both manipulations is the same. The only

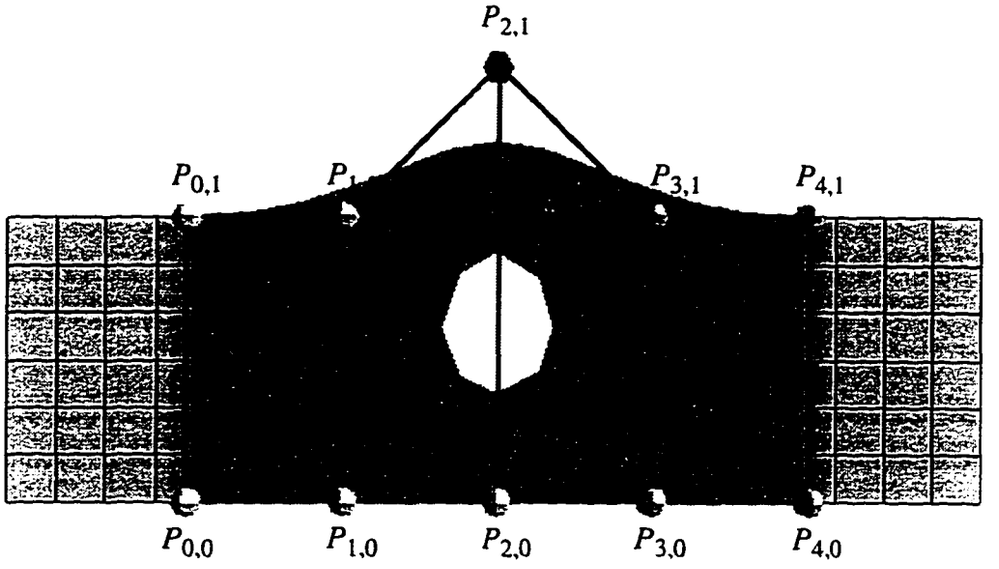


Figure 5.4 Local modification with continuous boundary

difference occurs in the computation of the design velocity field. The PinchGlove™, as shown in Figure 5.5, is used to specify the design variable after the design is parametrized, and to move the design variable to change the shape of the structure.

Note from Figure 5.5 that for the same amount of change in node N_{18}, the direct and indirect manipulations result in two different shapes. For the direct manipulation method, the pseudo-inverse method (Section 4.1.3) is used to calculate the minimum movement of all possible control points. In this case, the control points $P_{0,1}$, $P_{1,1}$, and $P_{2,1}$ all contribute to the movement of node N_{18}. For the indirect manipulation method, however, only the control point $P_{1,1}$ is changed thus the resulting shape change is different.

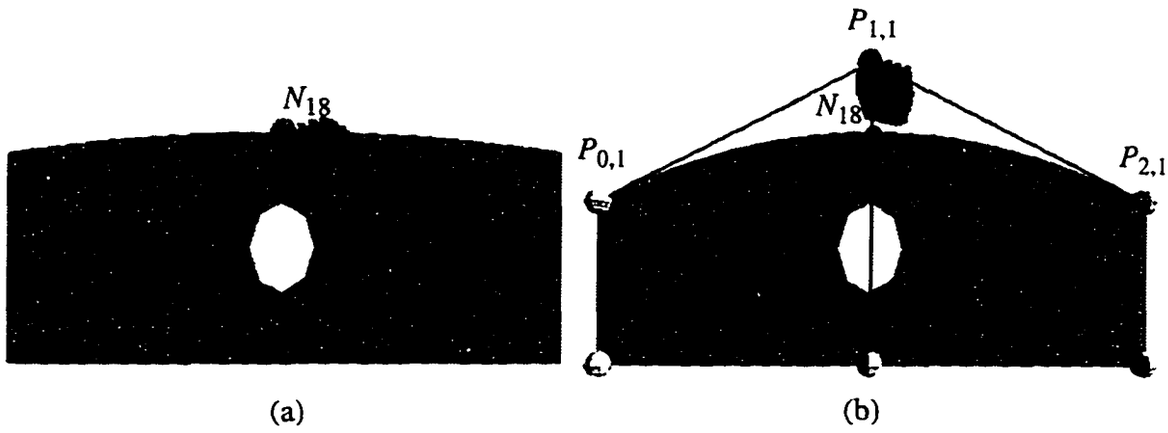


Figure 5.5: (a) Direct manipulation and (b) Indirect manipulation

5.4.3 Feature Preserving Technique

For the above example, the isoparametric mapping method is used to updated all of the finite element nodes embedded inside the design area. As we discussed in Section 5.2, the feature within the design area is not easy to maintain. Figures 5.3, 5.4 and 5.5 show that the round hole in the cantilever beam becomes an elliptic hole after the shape changes. To preserve the round hole, the designer can use the local modification technique to change the design area without embedding the round hole. As shown in Figure 5.6 (a), only the top portion of the beam is embedded inside the NFFD design area and the finite element nodes around the hole are excluded from the modification. Figure 5.6 (b) and (c) show the shape changes of the cantilever beam without changing the shape of the round hole. However, this method may violate the regularity and linear requirement of the overall finite element mesh resulting in an inaccurate linear sensitivity calculation.

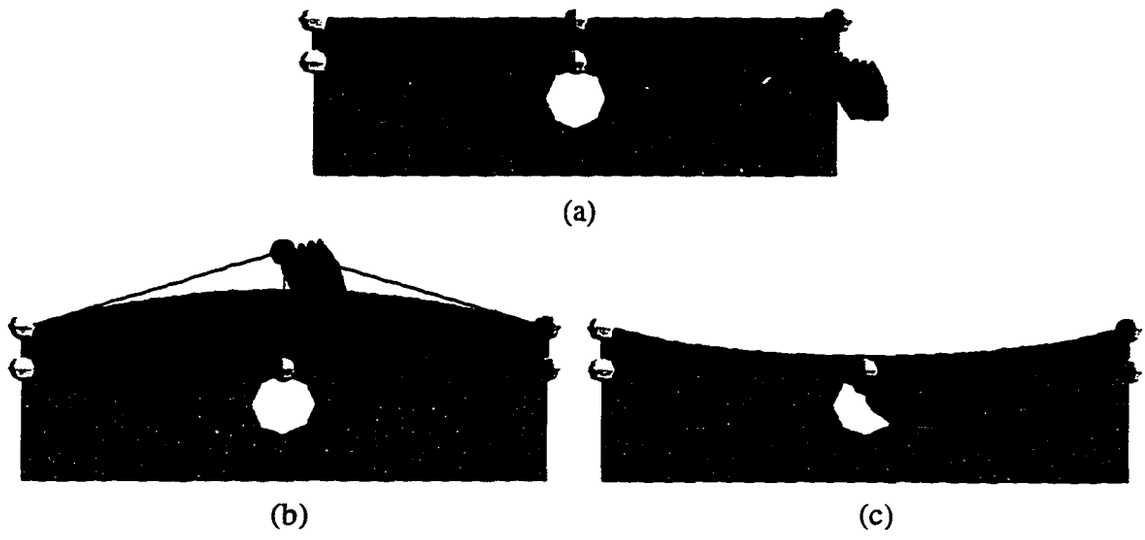


Figure 5.6: Local modification for feature preserving

Another approach uses the hybrid method which combines the isoparametric mapping method for the boundary DVF and the boundary element method for the domain DVF. To perform the hybrid method, the designer should identify the boundary nodes which will be constrained for preserving the feature. As shown in Figure 5.7 (a), the boundary nodes will be identified by small spheres if the hybrid method is used. The designer then specifies the constrained nodes by picking the corresponding spheres. The constrained spheres will be highlighted by changing the color as seen in Figure 5.7 (b). After the constrained node specification process, the rest of the boundary nodes are parametrized in order to perform the isoparametric mapping method, and a MSC/NASTRAN input data file is generated to calculate the domain DVF and the sensitivities of the structural responses. The MSC/NASTRAN data block DESVCP contains the DVF information for each finite element node, and only the interior unconstrained finite element nodes use the derivative information in DESVCP to

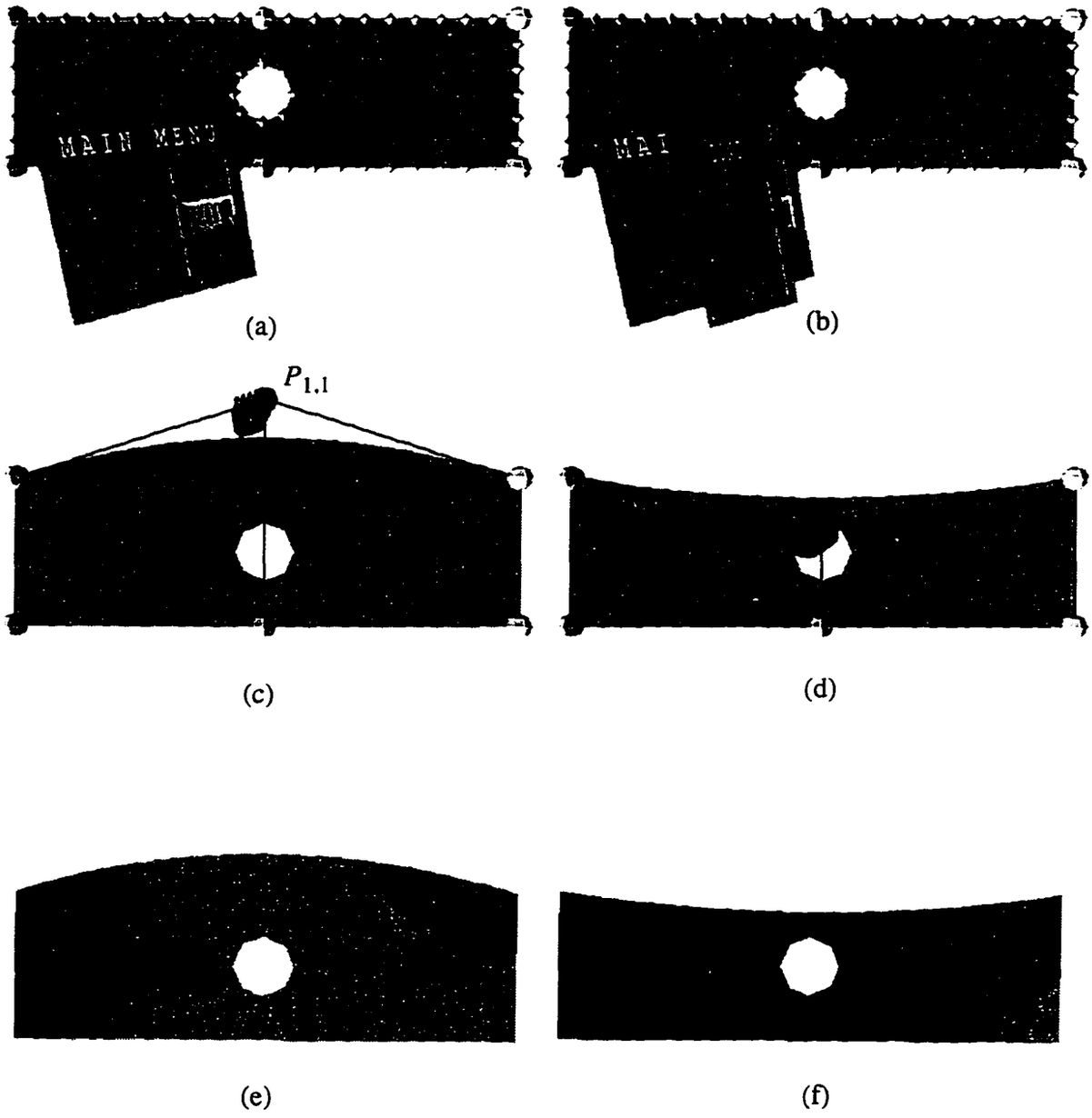


Figure 5.7: Hybrid method for feature preserving
(a) Press **CONT** button to bring up the boundary spheres
(b) Use the PinchGlove to specify the constraints
(c) Increase the design variable
(d) Decrease the design variable
(e) Analysis approximation of shape change in (c)
(f) Analysis approximation of shape change in (d)

update the mesh. Figure 5.7 (c) and (d) show the shape changes when the designer changes the design variable. The data block DSCM2 contains the sensitivities of the structural responses. The analysis models, as shown in Figure 5.7 (e) and (f), can thus update the results through linear Taylor series.

5.4.4 Reduced Basis Method

If the control points are selected as the design variable for the indirect manipulation, several control points can be moved as a group in order to reduce the number of the design variables or provide the versatility of the shape changes. It is often referred to the *reduced basis method.* The reduced basis method is used to constrain the movement of several control points such that those constrained control points become one independent design variable. The group of constrained control points can be separated as independent control points and dependent control points, and the relationship of these control points can be simply described as

$$\mathtt{DCP} = \mathbf{M} \cdot \mathtt{ICP} \tag{5.17}$$

where DCP and ICP are 1×3 vectors which represent the coordinates the dependent control points and the independent control points respectively, and $\mathbf{M}$ is a 3×3 matrix which describes the linear relationship of the dependent control points and the independent control points. Figure 5.8 illustrates the constrained movements of the control points and the corresponding matrix $\mathbf{M}$.

Note that if the reduced basis method is used to define the design variable, the Equation (5.1) to compute the DVF should be rewritten as:

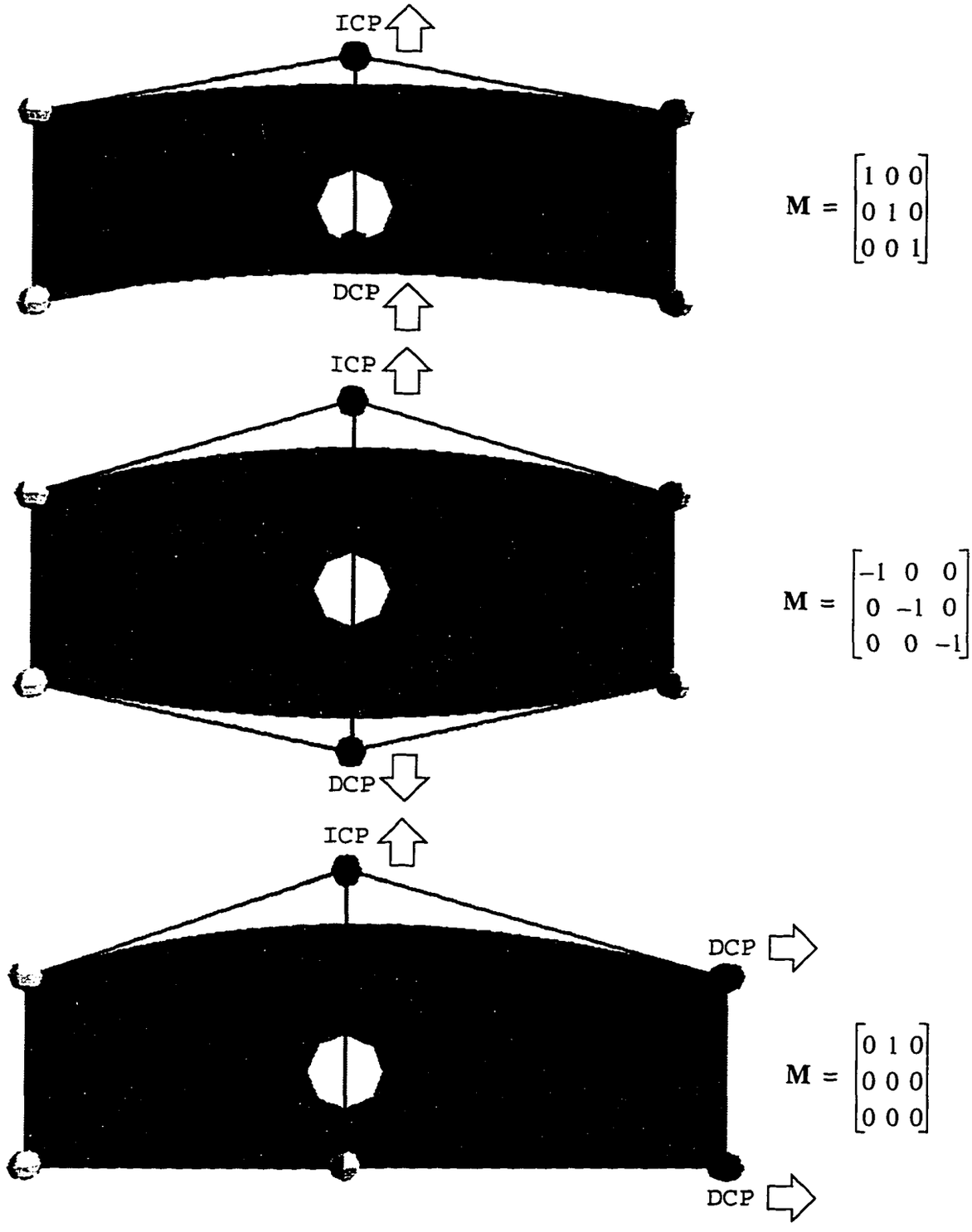


Figure 5.8: Various shape changes and the corresponding matrix **M** for the reduced basis method

$$V(S)_{3\times 1} = \sum_{i=1}^{n} \mathbf{M}_i \left[\frac{N_{x_i,p}(u) N_{y_i,q}(v) N_{z_i,r}(w) W_{x_i,y_i,z_i}}{Sum} \right]_{3\times 1} \qquad (5.18)$$

where n is the total number of the control points for the design variable, $N_{x_i,p}$, $N_{y_i,q}$, $N_{z_i,r}$ and W_{x_i,y_i,z_i} are the corresponding basis functions and the weighting factors of the control point P_{x_i,y_i,z_i} in the reduced basis, and $\mathbf{M}_i$ is the relationship matrix for the independent control point and the dependent control points as shown in Equation (5.17).

5.5 Numerical Examples

5.5.1 2D Cantilever Beam

5.5.1.1 Problem Definition

A 2D cantilever beam with a round hole is a typical example used to illustrate structural shape design sensitivity analysis [73][84][89][102]. In this section, it is again used to demonstrate the flexibility of the shape design and the accuracy of the sensitivity computation. The finite element model of the 2D cantilever beam, as shown in Figure 5.9, is clamped at the left end and a concentrated force of 10000.0 lb is applied at the top right corner. Young's modulus and Poisson's ratio are 3.0×10^7 psi and 0.3 respectively. The objective of this example is to explore a variety of shape designs in the virtual environment, and view the approximated analysis results in real time. The knot vectors which define the NURBS surface to embed the design model are $\mathbf{U} = \{0.0, 0.0, 0.0, 1.0, 1.0, 1.0\}$ and $\mathbf{V} = \{0.0, 0.0, 1.0, 1.0\}$.

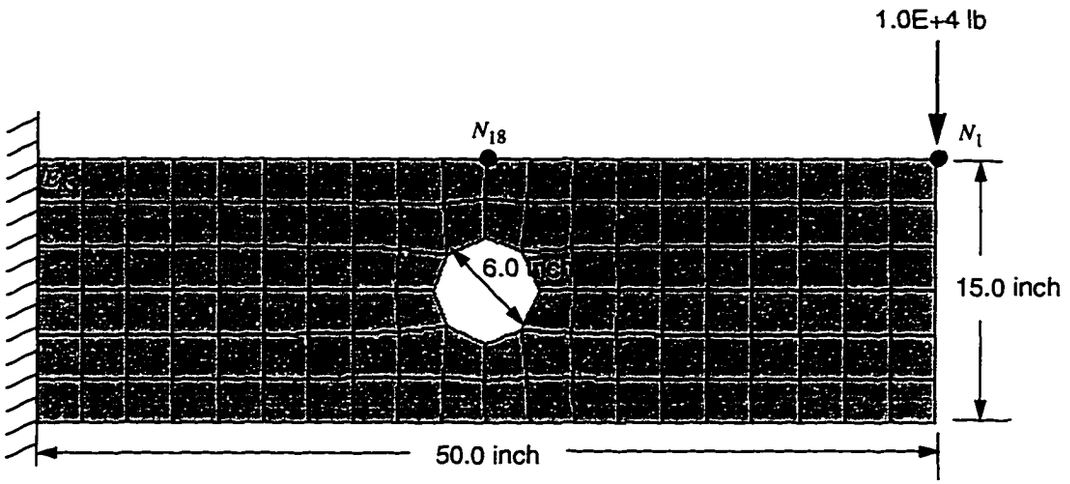


Figure 5.9: Finite element model of the 2D cantilever beam

5.5.1.2 Design Process

The design process of the 2D cantilever beam has been shown in Section 5.4. In this example, the cantilever beam model is all embedded inside the NURBS surface for the global modification. The control point $P_{1,1}$ in Figure 5.7 is selected as the design variable. The finite element nodes on the round hole are constrained and the hybrid method is used to determine the DVF. Once the sensitivity information is obtained, the designer can move the design variable and see the resulting changes of the analysis model in real time. The displacement and vonMises stress of the modified shape are calculated through linear Taylor series approximation and displayed on the analysis result model as shown in Figure 5.7 (e) and (f). The accuracy of the sensitivities is determined by the comparison of linear approximation and the finite element reanalysis.

5.5.1.3 Verification of Design Sensitivity

The nodal point N_1 and element E_{85}, as shown in Figure 5.9, contain the highest displacement and the highest element vonMises stress respectively. The finite element solution and the approximated solution of the displacement at node N_1 and the stress at element E_{85} are tabulated and illustrated in Tables 5.1 and 5.2. Note that the first columns in both tables are the movement of the control point $P_{1,1}$ in the y direction. A positive value for the design variable indicates that the control points are being pulled upward (positive y direction) as shown in Figure 5.7 (c), and the negative value represents the control point being pulled down (negative y direction) as in Figure 5.7 (d).

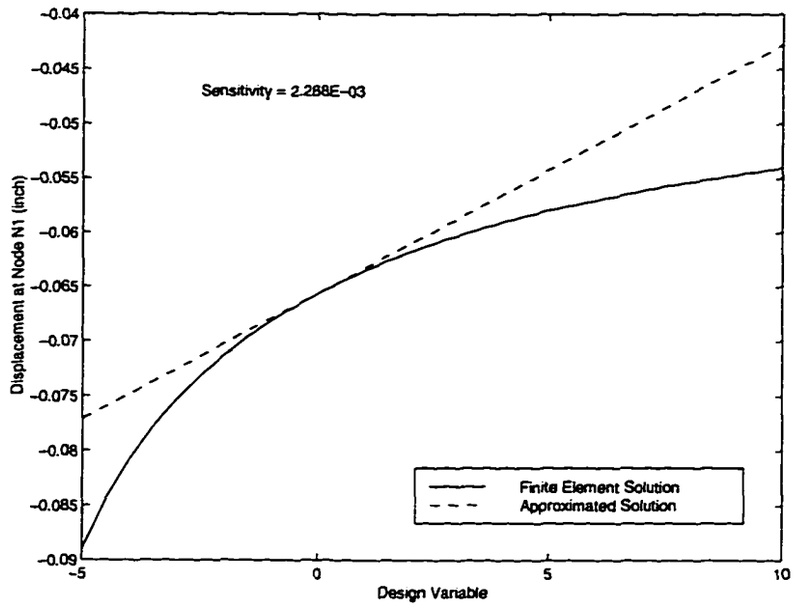
Figure 5.10 (a) and (b) shows the displacement of node N_1 in the y direction and the vonMises stress of element E_{85} respectively for both the approximation results and the exact solutions of the finite element reanalysis. In Figure 5.10 (a) and (b), the exact solutions of the displacement and the vonMises stress are shown to be nonlinear to the changes of the design variable, however, the first derivative derived from MSC/NASTRAN shows a reasonable accuracy to predict the structural responses through linear approximation. From Tables 5.1 and 5.2, the range of the design variable changes within ± 4.0 results in the error of displacement approximation less than 10.0% and the error of stress approximation less than 0.1%. Thus we agree that the sensitivity method presented in this chapter provides a good approximation for the linear static analysis of the cantilever beam shape design within a range of ± 4.0 design variable changes. The modified shape of ± 4.0 design variable changes are shown in Figure 5.11 (a) and (b) respectively.

Table 5.1. Verification of displacement at node N_1

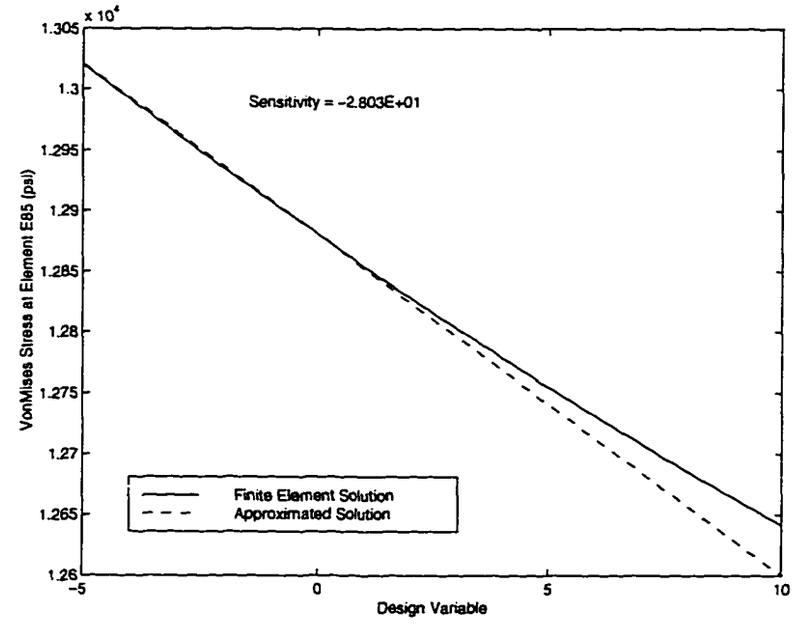
Movement of the Design Variable (inch)	Approximation (inch)	FEA reanalysis (inch)	Error %
5.0	-5.422E-02	-5.793E-02	6.404%
4.0	-5.650E-02	-5.905E-02	4.318%
3.0	-5.879E-02	-6.033E-02	2.553%
2.0	-6.108E-02	-6.182E-02	1.085%
1.0	-6.337E-02	-6.357E-02	0.315%
-1.0	-6.794E-02	-6.818E-02	0.352%
-2.0	-7.023E-02	-7.130E-02	1.500%
-3.0	-7.252E-02	-7.530E-02	3.692%
-4.0	-7.481E-02	-8.072E-02	7.322%
-5.0	-7.710E-02	-8.897E-02	13.342%

Table 5.2. Verification of vonMises stress at element E_{85}

Movement of the Design Variable (inch)	Approximation (psi)	FEA reanalysis (psi)	Error %
5.0	1.274E+04	1.275E+04	0.079%
4.0	1.277E+04	1.278E+04	0.078%
3.0	1.279E+04	1.280E+04	0.078%
2.0	1.282E+04	1.283E+04	0.077%
1.0	1.285E+04	1.285E+04	0.000%
-1.0	1.291E+04	1.291E+04	0.000%
-2.0	1.293E+04	1.293E+04	0.000%
-3.0	1.296E+04	1.296E+04	0.000%
-4.0	1.299E+04	1.299E+04	0.000%
-5.0	1.302E+04	1.302E+04	0.000%

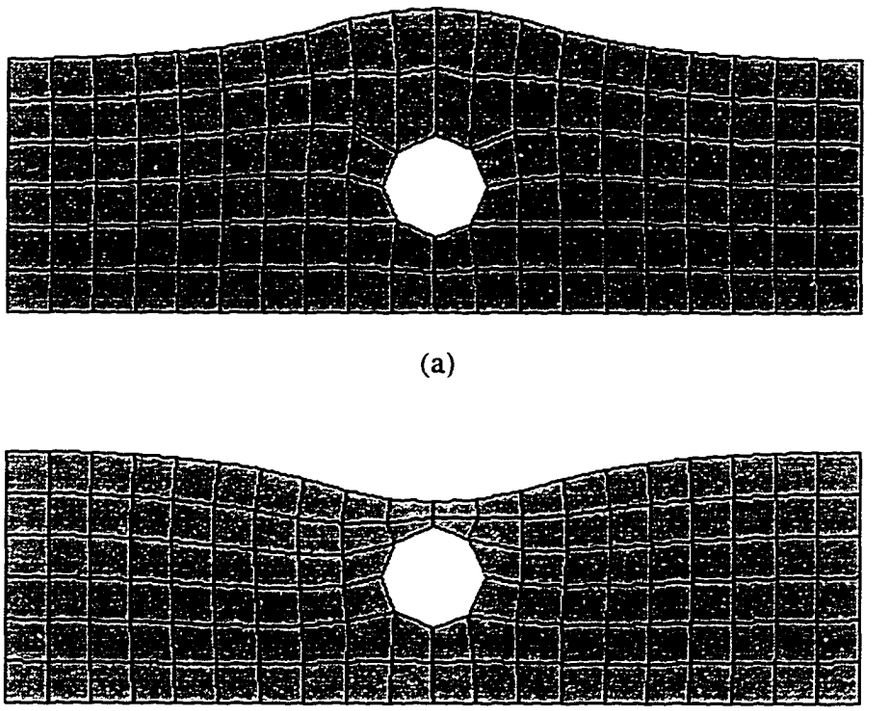


(a)



(b)

Figure 5.10: Comparison of the linear approximation and finite element reanalysis for
(a) Displacement at node N_1
(b) VonMises stress at element E_{85}



(a)

(b)

Figure 5.11: Modified 2D cantilever beam with
(a) Design variable +4.0 (b) Design variable -4.0

5.5.2 3D Engine Connecting Rod

5.5.2.1 Problem Definition

The connecting rod shown in Figure 5.12 (a) is a quarter model in order to take advantage of the rod's geometric symmetry. The dimensions of the rod are shown in Figure 5.12 (b). To simplify the problem, the boundary condition consists of a clamped condition in the crankshaft end and a uniform pressure of 100 MPa at the pin end to simulate the firing force.

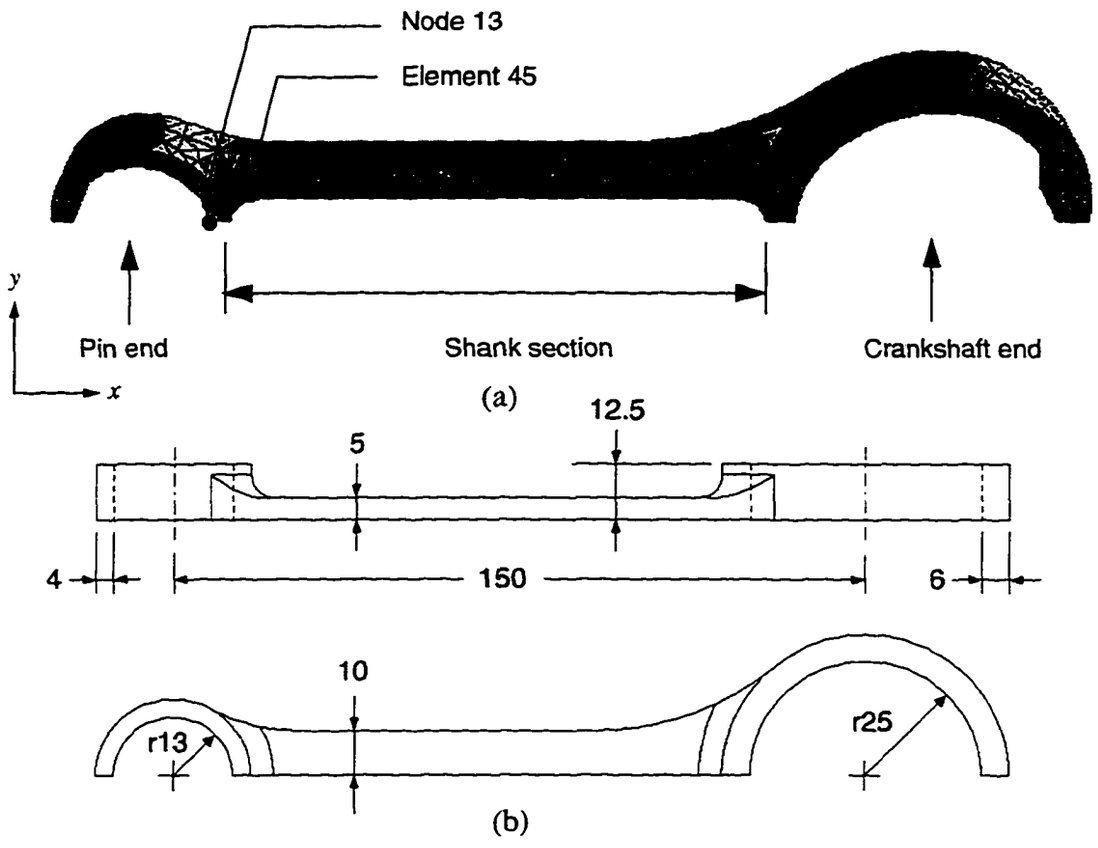


Figure 5.12 (a) Engineering connecting rod (b) Dimensions of the rod

Young's modulus and Poisson's ratio are 2.07×10^5 MPa and 0.3 respectively. The purpose of this connecting rod design is to explore a variety of different shapes in the shank section and determine the impact on the stresses and displacement.

5.5.2.2 Design Process

Since only the shank section of the connecting rod case is of interested, the control point lattice is constructed, using the PinchGlove, only around this area. A three-dimensional bounding box as shown in Figure 5.13 (a) is presented when the designer presses the button

FFD BOX on the virtual menu, and the bounding box can be resized to cover the design area using the PinchGlove. After the design area has been specified, the parametrization process is performed to determine the parametric value of the finite element grid points which are embedded inside the control point lattice. The color of the design area will be highlighted after the parametrization process is complete (see Figure 5.13 (b)). The designer can then choose either a control point or a grid point as the design variable. In this case, the control points $P_{1,1,0}$ and $P_{1,1,1}$ as shown in Figure 5.13 (b) are selected as one design variable. The DVF with respect to the selected design variable is calculated when the design variable is specified. Note that the selected control points are moved simultaneously in the same direction, thus the reduced basis method as presented in Section 5.4.4 should be used to determine the DVF. The DVF is then used in the input data file for MSC/NASTRAN sensitivity analysis solution sequence to determine the stress sensitivity information for the structural responses. When the sensitivity computation is complete, the sensitivity analysis results are brought into the virtual environment automatically for the designer to perform the what-if studies. The designer can change the design variable by moving the control point or the nodal point of the finite element model by using natural hand motions, and the finite element mesh and analysis results will be updated correspondingly through linear sensitivity approximation without performing a re-analysis. As shown in Figure 5.13 (c) and (d), the designer will be able to move the control point to change the shape of the connecting rod and also view the changes of the structural responses in real time. In Figure 5.13 (a) to (d), the top representation of the connecting rod is where the user interacts with the geometry. The resultant displacements and stresses are displayed in the lower representation of the connecting rod.

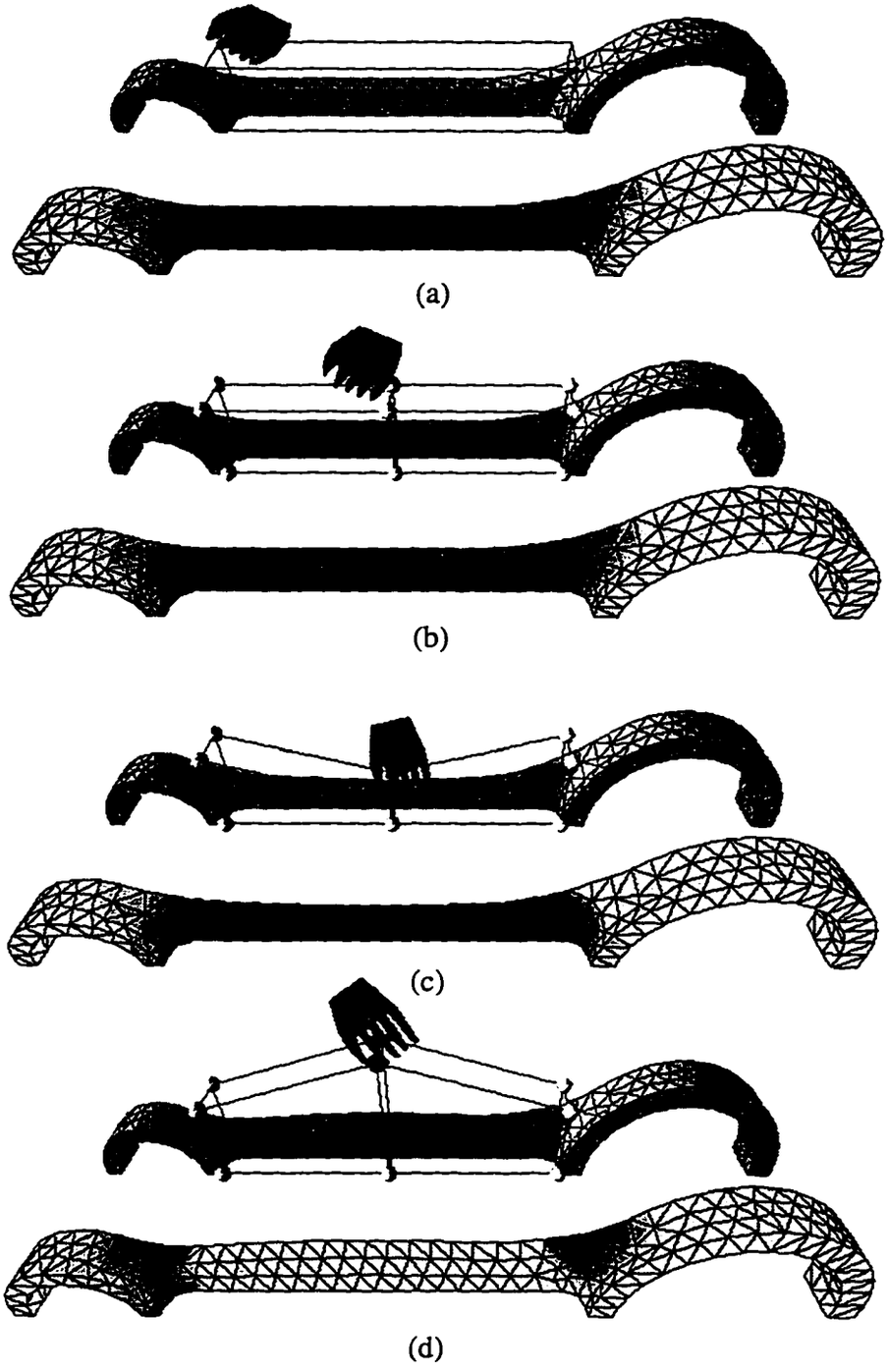


Figure 5.13: Process of the shape design
(a) Design area specification
(b) Parametrization and design variable selection
(c) Shape change (decreasing design variable)
(d) Shape change (increasing design variable)

5.5.2.3 Verification of Design Sensitivity

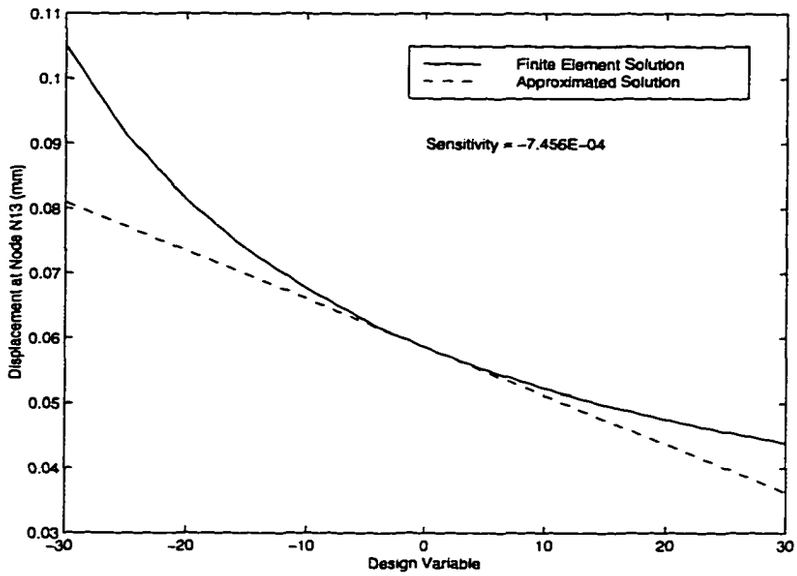
The accuracy of the linear sensitivity approximation of displacement and stress is verified by reanalyzing the modified connecting rod. The nodal point N_{13} and the element E_{45}, as shown in Figure 5.12 (a), contain the highest displacement and the highest element vonMises stress respectively. The finite element solution and the approximated solution of the displacement at node N_{13} and the stress at element E_{45} are tabulated and illustrated as in Tables 5.3, 5.4 and in Figure 5.14:

Table 5.3. Verification of displacement at node N_{13}

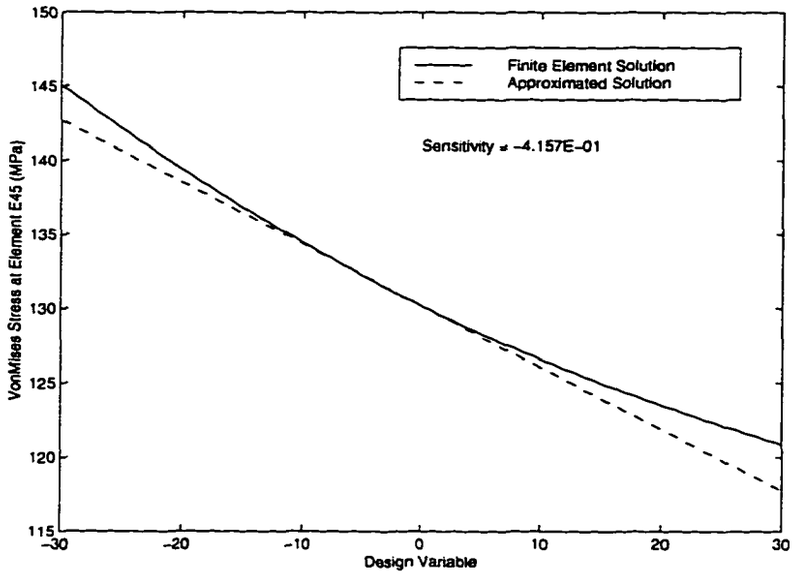
Movement of the Design Variable (mm)	Approximation (mm)	FEA (mm)	Error %
20.0	4.367E-02	4.747E-02	8.005%
10.0	5.112E-02	5.221E-02	2.088%
5.0	5.485E-02	5.514E-02	0.526%
-5.0	6.231E-02	6.266E-02	0.558%
-10.0	6.603E-02	6.758E-02	2.294%
-20.0	7.349E-02	8.127E-02	9.573%

Table 5.4. Verification of displacement at element E_{45}

Movement of the Design Variable (mm)	Approximation (MPa)	FEA (MPa)	Error %
20.0	1.219E+02	1.235E+02	1.296%
10.0	1.261E+02	1.266E+02	0.345%
5.0	1.281E+02	1.283E+02	0.156%
-5.0	1.322E+02	1.322E+02	0.000%
-10.0	1.343E+02	1.344E+02	0.001%
-20.0	1.385E+02	1.394E+02	0.646%



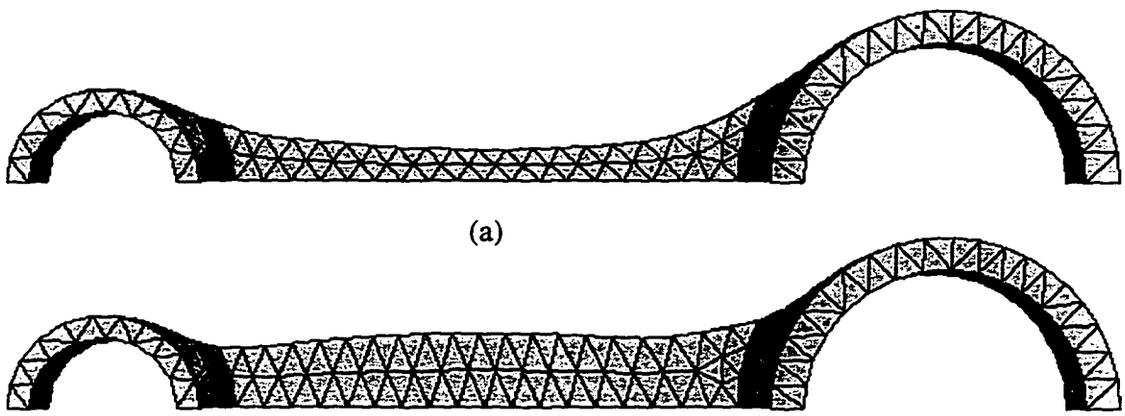
(a)



(b)

Figure 5.14: Comparison of finite element reanalysis results and approximated results for
(a) Displacement at node N_{13}
(b) VonMises stress at element E_{45}

Figure 5.14 (a) and (b) show the displacement of node N_{13} in the x direction and the von-Mises stress of element E_{45} respectively for both the approximation results and the exact solutions. A positive value for the design variable indicates that the control points are being pulled upward (positive y direction) as shown in Figure 5.13 (d), and the negative value represents the control point being pulled downward (negative y direction) as in Figure 5.13 (c). The exact solutions of the displacement and the stress are nonlinear to the changes of the design variable, however, the linear approximations show reasonable accuracy for small changes of the structural shape. From Tables 5.3 and 5.4, the range of the design variable changes within ± 20.0 results in the error of displacement approximation less than 10.0% and the error of stress approximation less than 1.3%. Thus we agree that the sensitivity method presented in this paper provides a good approximation for the linear static analysis of the engine connecting rod shape design. The modified shapes of ± 20.0 design variable changes are shown in Figure 5.15 (a) and (b) respectively.



(a)

(b)

Figure 5.15: Modified engine connecting rod with
(a) Design variable -20.0
(b) Design variable +20.0

5.6 Conclusion and Future Developments

Sensitivity methods, finite element analysis, and free-form deformation have been combined within a virtual environment to facilitate structural shape design. NURBS-based free-form deformation (NFFD) methods and direct manipulation techniques are used as the interface between the VR interaction and the finite element mesh changes. The addition of analysis results to the virtual environment provides further information to the designer which helps to guide the choice of design changes. The incorporation of sensitivities which are used to approximate the analysis results facilitates real time interaction and what-if studies which further enhance the design change, thus it helps the designer to achieve a viable, optimal solution in a timely manner.

A 2D cantilever beam with a round hole and a 3D engine connecting rod have been used to demonstrate the feasibility and accuracy of sensitivity-based structural shape design in the virtual environment. The designer has various options of direct or indirect manipulation, global or local modification, feature preserving techniques, and a reduced basis method to explore a variety of structural shape changes and view the structural responses which are calculated using linear approximation.

One limitation of the NFFD method applied in this research is the use of a rectangular bounding box to define the NURBS surface or volume. Thus the structure embedded inside the bounding box should be relatively rectangular in shape in order to maintain the regularity of the finite element shape after deformation. For more complicated shape design, multiple bounding boxes should be used to embed the structure such that each section of the structure embedded inside one bounding box will be relatively rectangular in shape. Future develop-

ment will include the use of multiple bounding boxes for complicated shape design, and the investigation of using Extended Free-Form Deformation (EFFD) [54] such that the bounding box can be arbitrarily shaped to allow for more variety in shape design.

Another limitation of the NFFD method is that the shape of the structure where the boundary condition is applied can not be changed, since the sensitivity computations for the structural responses assumes that the boundary conditions remain the same. If the boundary condition changes, the derived sensitivities will not be accurate. For example, if the highlighted control points as shown in Figure 5.16 move simultaneously, they can be considered as one design variable which can change the width of the connecting rod. However, the boundary conditions on the pin end and the crank shaft end are also changed. Although the modified finite element mesh still maintains the linearity and regularity with respect to the change of the design variable, the sensitivities of the displacement and the stress are not correct. Figure 5.17 (a) and (b) shows the comparison plots of the linear approximation and the finite element reanalysis of the displacement at node N_{13} and element E_{45}. The sensitivities obviously can not be used in the linear approximation to update the structural responses.

Another area of future development is the method used to specify a constraint boundary. Since the finite element model does not provide information regarding feature definition, it is difficult to specify constraints for feature preserving. For example, the finite element model of the cantilever beam does not tell us which nodes are on the boundary of the round hole. The designer has to specify the constraints by selecting the finite element nodes on the round hole manually. For a large model with a considerable number of finite element nodes on the constrained feature, the process to select the constrained nodes can be very inefficient. One way to

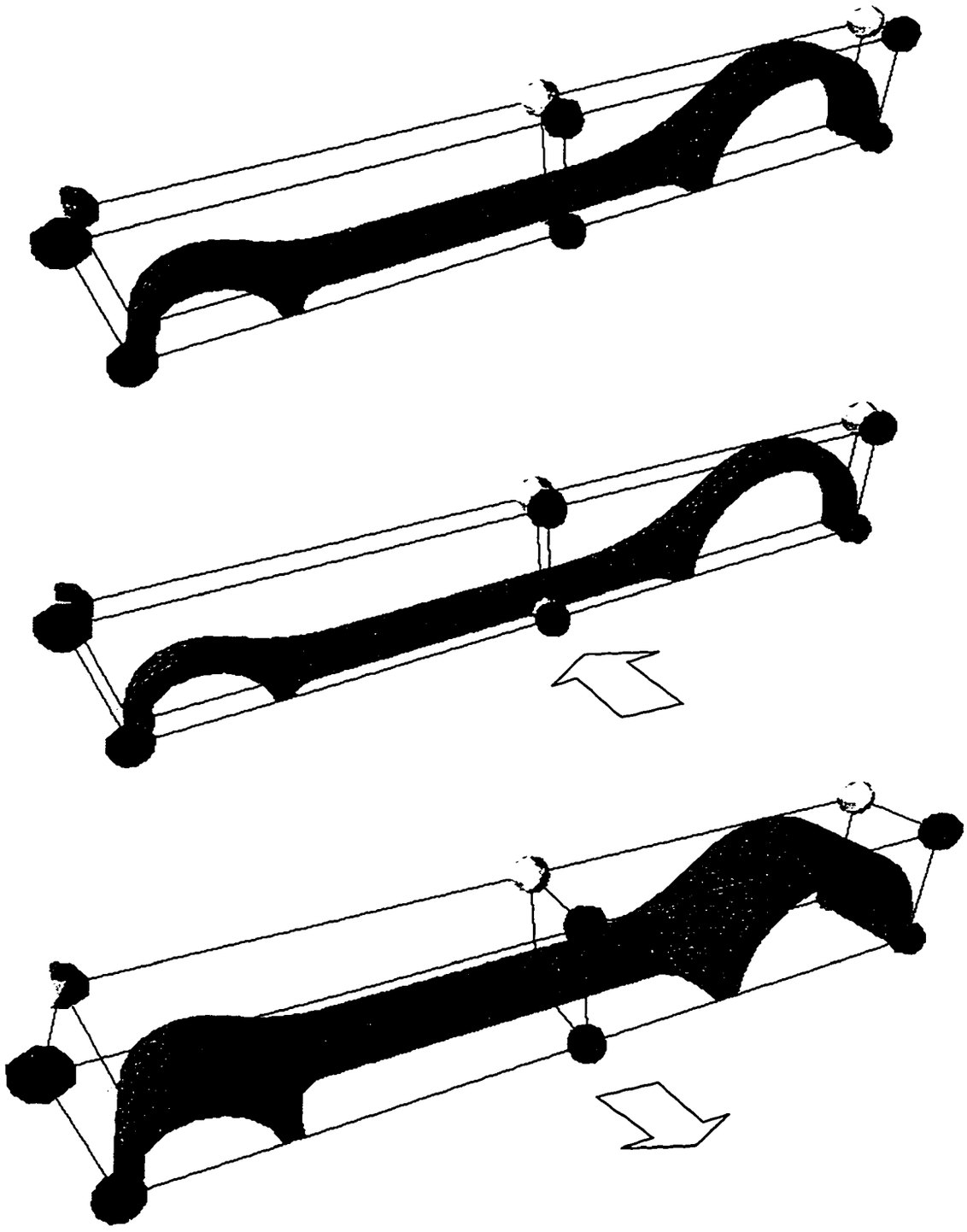
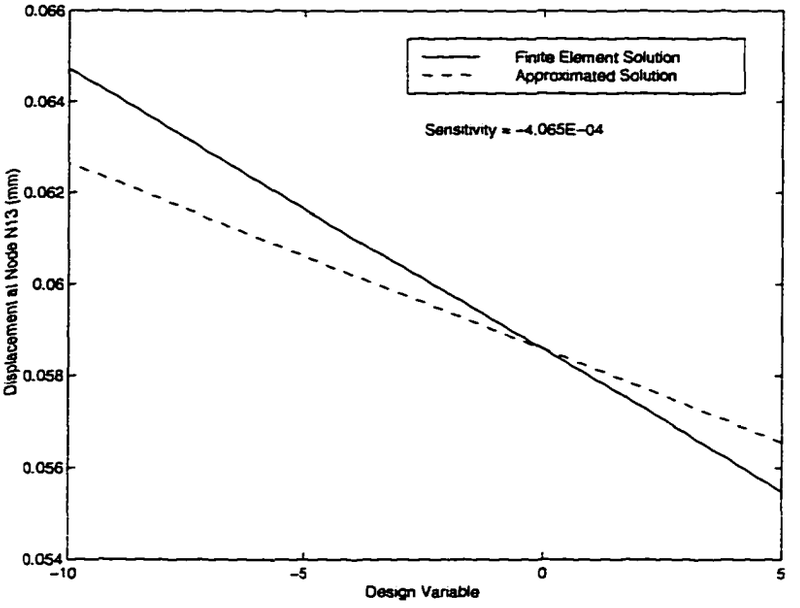
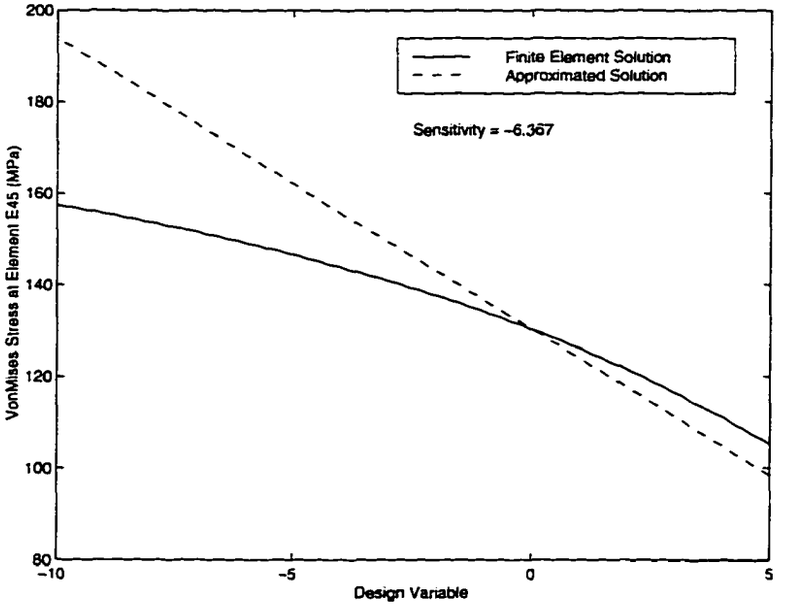


Figure 5.16: One example of choosing a design variable which will change the boundary conditions



(a)



(b)

Figure 5.17: Invalid sensitivity approximation for
(a) Displacement at node N_{13}
(b) VonMises stress at element E_{45}

overcome the selection process is to take advantage of virtual reality devices and use the data glove to specify the constrained feature by sweeping the constraint area. Another method could interface with a CAD data file to provide information about the geometric feature. For example, the IGES file format [50] contains the B-spline surfaces (entity 128) and curves (entity 126) to represent the boundary of the geometry model. The cantilever beam with a round hole, for example, can be represented by 5 B-spline curves for the boundary of the beam and the round hole, and 1 B-spline surface to be trimmed by the B-spline curves. The designer can thus select the round hole through the data file to be constrained during the shape change. An inverse mapping technique, as we discussed in Section 4.1.2, can be used to find the finite element nodes located on the curve of the round hole thus the designer does not have to specify the finite element nodes individually to constrain the hole and the efficiency of preserving the features of the geometry model can be improved.

Furthermore, the approximation method used in this research is based on linear Taylor series. For wider ranges of shape changes, the linear approximation may result in higher inaccuracy. One method to improve the approximation accuracy is by using a curve fitting technique [103] to interpolate the analysis results between two exact solutions. If the higher order derivative information can be obtained, high order polynomial equations [43][103][104] can also be used to calculate the approximated solutions.

CHAPTER 6. INTERACTIVE DESIGN OPTIMIZATION

Optimization is a design process concerned with achieving the best performance of a given design while satisfying certain restrictions or constraints. The optimization algorithm for most structural shape design is typically set up as a *black box* operation. The desired design characteristics, the design parameters, and the design constraints are entered into the box, and the resultant design comes out with a minimum or maximum solution that satisfies all the constraints. This kind of optimization procedure requires the designer's prior knowledge to be quantified and applied in the form of objective functions and constraints before the optimization process begins. However, this approach may suffer from two drawbacks. First, if the designer is not pleased with the outcome of this optimal design, a redesign process is performed such that the parameters in the objective function or constraints must be adjusted and then the optimization process must restart. For a large problem with a complicated structure, this redesign process is tedious and time consuming for setting up proper parameters and constraints to approach an optimal design, resulting in long lead time for new, innovative products. Second, it requires that the designer should identify proper design variables and their upper and lower bounds, all important constraints, and the objective function precisely for the structural optimization algorithm. However, in most practical applications, trade-offs between the constraints and the optimal design exist, in fact some constraints may become more or less important that the designer did not foresee in the initial attempt to define the problem. The basis of the approach presented here is the belief that giving the designer an interactive design tool, which can be used to refine the formulation of the optimization problem and obtain the

optimal results through fast approximations, will foster creativity and encourage some of the *what-if* questions to be explored to proceed toward an optimum design among all trade-off competitors.

The interactive design optimization presented in this chapter provides a means for the designer to play an active role in structural design optimization to overcome the disadvantage of the traditional optimization process. The interactive design optimization technique will allow the designer to interactively modify the parameters and constraints, and explore a variety of optimal designs based on the designer's prior knowledge of this design problem. Being able to investigate design changes interactively will give the designer an excellent feel for the effect such changes have on the performance of the structural system.

The virtual environment can be considered as an advanced human-machine interface for highly immersive, real-time interaction with the computer data. The advantage of using VR techniques for the sensitivity analysis of structural shape design has already been presented in Chapter 5. VR can also enhance the interactive design optimization for structural shape design problems, since VR provides an intuitive way to specify the design variable and interact with the design model by using natural hand motion. It also provides a unique method to view the effect of analysis results with the stereo view and the head controlled viewpoint. The only challenge right now is to find a fast optimization method to obtain the optimal solution in a shorter time, even for a large computer model, in order to preserve the nature of real-time interaction in the virtual environment. The formulation of the interactive design optimization presented in this chapter is based on the linear approximation of the structural responses so a computationally intensive reanalysis can be avoided. Combined with a simple concave objec-

tive function, it can be easy to converge to a solution for the optimization problem. This approach allows the real-time interaction in the virtual environment.

In the following section, the traditional finite-element based structural optimization will be introduced. The interactive optimization technique applied in this research will also be discussed. Two examples using the interactive optimization technique in a virtual environment are presented to demonstrate the advantages of coupling the optimization process with interactive VR techniques for the structural size and shape design optimization.

6.1 Finite Element Based Structure Design Optimization

Finite element based design optimization techniques have emerged as a useful design tool for finding the optimal design for structural systems. Early structural design optimization was focused on minimizing the weight of an object using a finite element model made up of bars, membranes, beams, shells, plates, etc. where the transverse area or thickness of the elements is treated as the design variable, and the overall shape of the structure remains during the design process. These approaches are generally referred to as *size optimization*. Shape optimization is more complicated than size optimization. Since the boundary of the finite element model is continuously changing during the design process, careful consideration should be given to update an adequate finite element mesh and to obtain the sensitivity information. However, shape optimization has been proven to be a more accurate and more practical approach in many applications [66].

Shape design optimization has become more and more popular in the field of structural analysis implying that the importance of the structural shape to the performance of the struc-

ture has been recognized [65]. One of the first shape optimization techniques was presented by Zienkiewicz and Campbell [67] in 1973. In their approach, the boundary nodes served as the design variables, and the objective was to find an optimal shape of a fillet which reduces the stress concentration. Following their work, many researches have focused on shape design problems. With various choices of design variables as discussed in Section 5.1, more variety of shape changes can be achieved, and with continuous shape boundary maintained, the optimal shape becomes practical for physical product manufacturing. Many practical applications for structural shape design optimization have been investigated, including the design of an engine connecting rod [84][89][102][105], an engineering bearing cap [79][89], a control arm [82][87], and a turbine blade [83]. More literature on structural shape optimization can also be found in references [65][66] and [97].

Despite the various applications for the shape design optimization problem, the general mathematical form for the optimization process can be described as

$$\begin{aligned} &\textbf{\textit{minimize}} && f(S) && \\ &\textbf{\textit{subject to}} && g_i(S) \le 0 && i = 1, NIC \\ &&& h_j(S) = 0 && j = 1, NEC \\ &&& S^L \le S \le S^U && \end{aligned} \tag{6.1}$$

where f is the objective function of the structural characteristics to be minimized with respect to the shape design variable vector S. Typically, the objective function can be the volume or weight of the overall structure to reduce the size of the structure, the maximum vonMises stress or tangential stress to decrease the stress concentration, or the displacement of a specific area to prevent interference with other components. *NIC* is the number of the inequality con-

straints g_i, and NEC is the number of the equality constraints h_j. S^L and S^U are the lower bound and upper bounds of the design variable vector respectively. The objective function and constraint functions are in most cases highly nonlinear with respect to the shape design variable, and the number of design variables and constraint functions are often large for complicated design problems such that it is difficult to obtain the optimal solution through analytical methods. Thus the solution of Equation (6.1) is generally subject to numerical optimization techniques. Detail information about the numerical optimization method is beyond the scope of this chapter. A variety of numerical optimization methods can be found in references [64] and [83].

Currently, the interaction a designer has with the optimization process is limited to the initial stage of problem definition, where the designer specifies an objective function, design parameters, and constraints. The numerical software is used to search for an optimal solution iteratively until the solution satisfies all the constraints. The interactive design optimization method presented in this chapter provide a different approach in the design optimization process such that the designer can interactively modify the design definition and obtain the optimal solution in an efficient way. A key to this approach, which is referred to as *interactive design optimization*, is to use the designer-weighted cost functions with soft constraints. Detail information will be introduced in the following section.

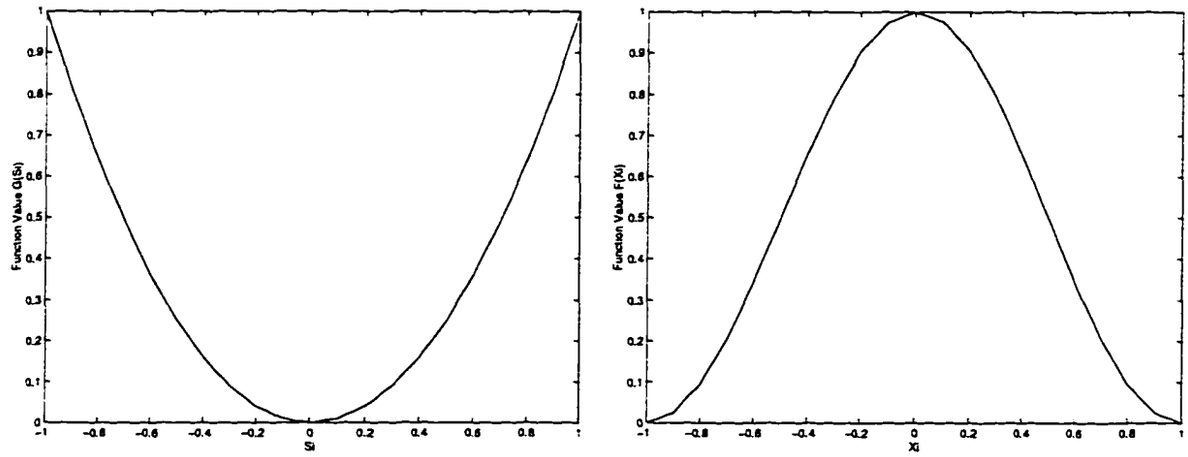
6.2 Interactive Design Optimization

The concept of the interactive design optimization is mathematically similar to the Sequential Unconstrained Minimization Techniques (SUMT) [106]. SUMT methods are pop-

ular for the solution of multivariable constrained optimization problems. The constrained functions are converted into the objective function using the penalty function concepts to cast the original constrained minimization into an unconstrained problem. By solving the unconstrained problem repeatedly with weighting factors of the penalty terms increasing or decreasing, the solution of the unconstrained function can approach the minimum solution of the original constraint problem. Such constraints are often referred to as *hard constraint* which allows no constraint violation. Starkey and Bernard proposed the designer-weighted objective function with soft constraints such that the constraints are not rigidly defined. Rather the designer would like to explore the effects of relaxing the strict constraints and see the effects on the overall performance of the objective function [107]. The concept of soft constraints can be explained through the cost function C in the following mathematical form

$$C(S, X) = \sum_{i=1}^{k} A_i G(S_i) + \sum_{j=1}^{l} B_j F_j(X_j) \tag{6.2}$$

where G is the *size-of-change* constraint function which becomes large when the design variable S_i begins to exceed prescribed limits. k is the number of the shape design variables and l is the number of selected structural responses, respectively. A_i and B_j are the designer weighted parameters (DWPs) such that the designer can interactively change DWPs to determine the relative significance of each term in Equation (6.2). In references [43] and [107], this function was generally set up as a quadratic function as shown in Figure 6.1. The function in Figure 6.1 (a) suggests that the current design is a relatively *good* design, but it allows the designer to explore the neighboring design variables and see if there is any possible



(a) (b)

Figure 6.1: Typical penalty functions

improvement for the current design, thus when the size of the design change begins to exceed the prescribed limits, the cost function G becomes large. In Figure 6.1 (b), F_j is a continuous function that produces a large number for undesirable structural response X_j, which can be displacement, stress, or structural frequency. Note that X_j is also a function of the design variable vector $\boldsymbol{S}$, and the value of X_j can be approximated by the sensitivity information of $\partial X_j / \partial S_i$. Since the cost function can be simplified to a quadratic function of the design variable $\boldsymbol{S}$, it is easier to find the minimum solution of Equation (6.2) than the original optimization problem definition in Equation (6.1). Thus the interactive design in the virtual environment to obtain the optimal solution in a timely manner can be achieved.

6.3 Interactive Design Optimization in a Virtual Environment

In the following sections, two practical examples of applications for the interactive design optimization are presented in detail. The first example is a simple cantilever beam with size optimization objectives and vibration optimization objectives, and the second example is the 3D engine connecting rod where the trade-offs between the displacements and the von-Mises stresses are taken into consideration for the optimal shape design.

6.3.1 Simple Cantilever Beam Example

6.3.1.1 Problem Definition

A simple cantilever beam is shown in Figure 6.2 and the corresponding dimensions and its material properties are listed in Table 6.1. The cantilever beam can be represented as a lumped spring-mass system as in Figure 6.3 to simplify the analysis. The heights of all sections are chosen as design variables since the masses and the stiffness of the springs are functions of the heights, and are shown in Equations (6.3) and (6.4) respectively. The natural frequencies of the cantilever beam are of interest in this example. The first natural frequency is 102.71 Hz. The objective of the interactive design is to find the minimum changes of current weight and move the first natural frequency out of the undesirable frequency band which is 100 Hz ~ 110 Hz.

$$m_i = \rho \times W \times \frac{L}{12} \times h_i = 0.57 \ \ lb \tag{6.3}$$

$$k_i = \frac{12.0 \times W \times h_i \times E}{L} = 1.5 \times 10^7 \ \ lb/in \tag{6.4}$$

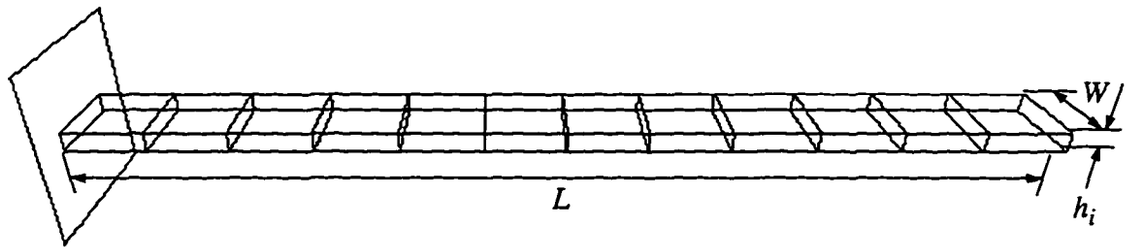


Figure 6.2: Cantilever beam with 12 finite elements

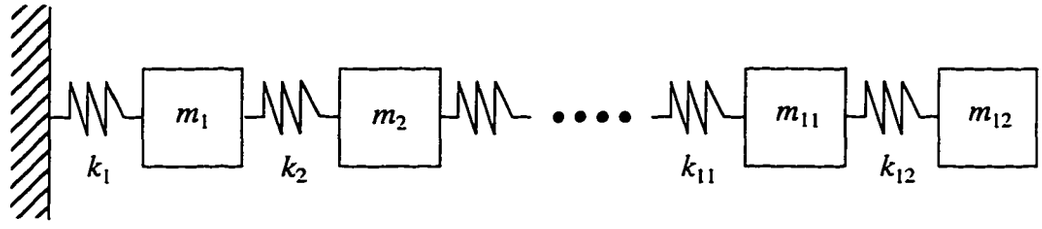


Figure 6.3: Lumped spring-mass system

Table 6.1: Dimensions and material properties of the cantilever beam

Length L	24.0 inch
Width W	2.0 inch
Height h_i	0.5 inch
Density ρ	0.284 lb/in^3
Young's Modulus E	3.0×10^7

6.3.1.2 Sensitivity Computation

In order to perform the interactive design optimization, the structural response sensitivities should be obtained first. In this example, the structural responses are the natural frequencies and modes shapes. For the lumped spring-mass system, the natural frequencies and mode

shapes are derived from the form

$$\mathbf{K}(\boldsymbol{h})U(\boldsymbol{h}) = \lambda\mathbf{M}(\boldsymbol{h})U(\boldsymbol{h}) \tag{6.5}$$

where the stiffness matrix **K**, the mass matrix **M**, the eigenvalues or natural frequencies λ, and the n-length eigenvectors or mode shapes U, are functions of the design variable vector $\boldsymbol{h}$, where $\boldsymbol{h} = \{h_i\}$, $i \in [1, 12]$. The first derivative of the ith eigenvalue, λ_i', and the jth eigenvector, U_j', with respect to the design variable $\boldsymbol{h}$ are given by Equations (6.6) and (6.7) [108].

$$\lambda_i' = \frac{U_i^T(\mathbf{K}' - \lambda_i\mathbf{M}')U_i}{U_i^T\mathbf{M}U_i} \tag{6.6}$$

$$U_j' = \sum_{k=1}^{n} a_k U_k^T \tag{6.7}$$

where $$a_k = \frac{U_k^T(\mathbf{K}' - \lambda_j\mathbf{M}')U_j}{\lambda_j - \lambda_k} \qquad \text{for } 1 \le k \le n \quad \text{and } k \ne j.$$

and $$a_j = -\frac{U_j^T(\mathbf{M}')U_j}{2}$$

6.3.1.3 Formulation of the Interactive Design Optimization

In order to obtain the minimum changes of the current design with the first natural frequency being moved out of the unwanted frequency band, the cost function to describe the problem definition is expressed as follows:

$$\textbf{\textit{minimize}} \qquad C(\boldsymbol{h}, \lambda) = A \cdot f(\boldsymbol{h}) + B \cdot g(\lambda_1) \tag{6.8}$$

$$f(\boldsymbol{h}) = \sum_{i=1}^{12} (\Delta h_i)^2$$

$$g(\lambda_1) = 1 - \cos(2\pi(\lambda_1 - u)/(l - u)), \qquad (l < \lambda_1 < u)$$

where l and u are the lower and upper bounds of the unwanted frequency band. Δh_i is the change of the design variable which is the height of each element. λ_1 is the first natural frequency and can be obtained through the linear Taylor series of

$$\lambda_1 = \lambda_{1_0} + \sum_{i=1}^{12} \left(\frac{\partial \lambda_1}{\partial h_i}\right) \Delta h_i \tag{6.9}$$

and λ_{1_0} is the original first natural frequency. The software used to search the optimal solution of Equation (6.8) is again CFSQP [61]. A in Equation (6.8) indicates the importance of the weight changes of the cantilever beam since the weight of the beam is directly associated with the design variable changes. As the design variable A increases, the weight change of the cantilever beam will be reduced. B in Equation (6.8) is the tolerance of the constraint which can be violated to explore the trade-off design between the constrained frequency and the weight changes. As B increases, the natural frequency will tend to move out of the unwanted frequency band.

6.3.1.4 Results

A virtual environment shown in Figure 6.4 contains the model of the cantilever beam, a CyberGlove, and a virtual menu system for the interactive design optimization. The designer

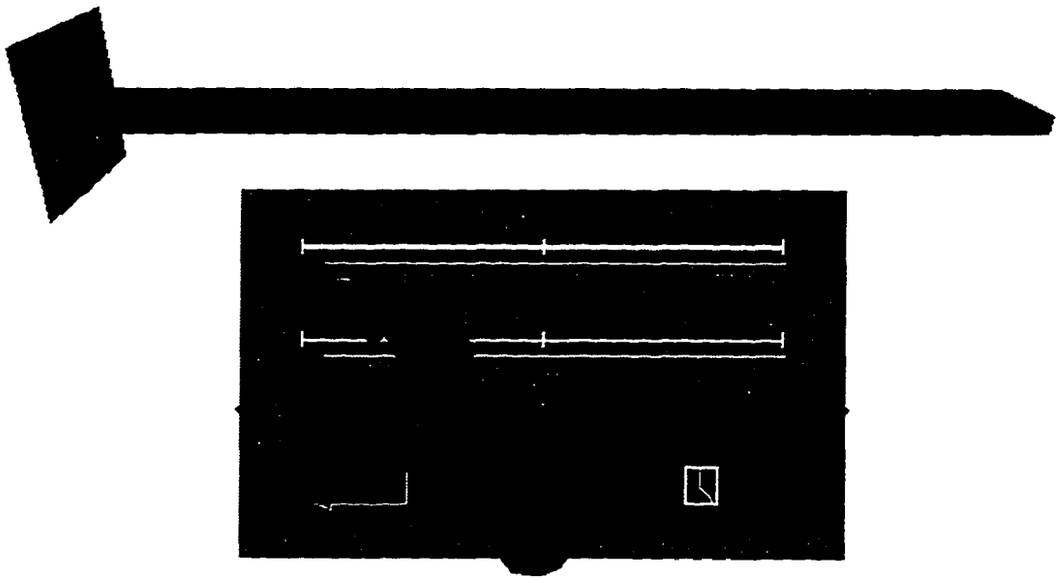


Figure 6.4: Virtual environment for cantilever beam size design optimization

can interactively change the DWPs A and B by moving the sliders on the virtual menu. By pressing the button **ENTER**, the optimization process is performed to find the optimal solution of Equation (6.8). The optimal shape of the cantilever beam as well as the optimal result of the natural frequency will be displayed after the optimal solution is obtained, and the designer can decide whether to keep this optimal design or perform another design based on the visual feedback of the virtual object.

Note that the unwanted frequency band in this problem is 100 ~ 110 Hz. In order to obtain the optimal solution with the frequency out of the unwanted band, the upper and lower bounds in Equation (6.8) should be carefully decided since it may affect the optimal solution

significantly. Normally, the upper bound u should be greater than 110, and the lower bound l should be less than 100 to order to compensate for the first term on the right-hand side of Equation (6.8), otherwise, the optimal solution of the natural frequency will never move out of the unwanted band. Here we define $l = 80$ Hz and $u = 120$ Hz. Figure 6.5 shows the plot of the optimal solution of the first natural frequency with respect to the design variable changes and the corresponding exact solution through the use of an eigenvalue solver. The numerical errors are listed in Table 6.2.

Table 6.2 shows a reasonable accuracy of the optimal solution obtained from Equation (6.8) with respect to the exact solution from the eigenvalue reanalysis. All the optimal solutions are under 3.0 percent error. Increasing the design parameter A tends to decreasing the approximation error since the changes of the heights are decreasing. Changing the design parameter B has less influence on the approximation error, since the design variable B only

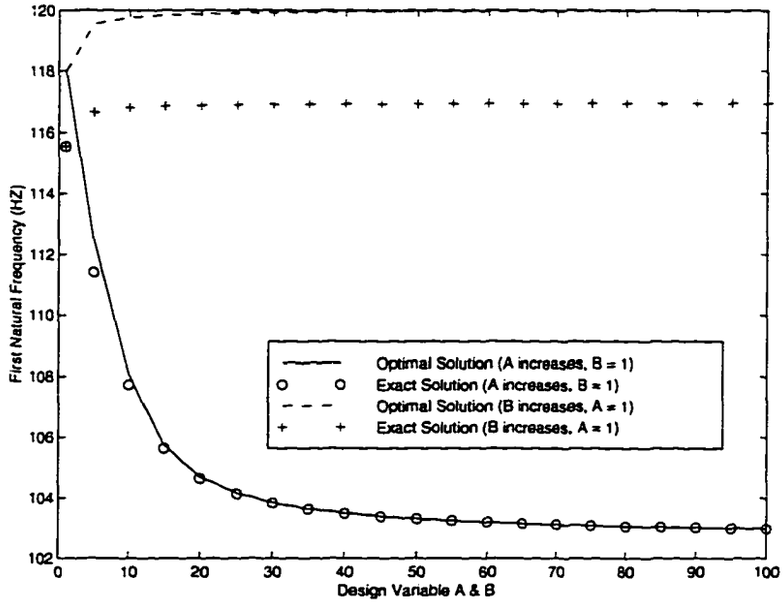


Figure 6.5: Comparison plot of the optimal solutions and the exact solutions for $l = 80$ Hz and $u = 120$ Hz

Table 6.2: Numerical errors of the approximated optimal solution

Design Variable A	Design Variable B	Optimal Solution (HZ)	Exact Solution (HZ)	Error %
1	1	117.99	115.54	2.12%
20	1	104.69	104.64	0.05%
40	1	103.49	103.49	0.01%
60	1	103.19	103.19	0.00%
80	1	103.06	103.06	0.00%
100	1	102.98	102.98	0.00%
1	20	119.89	116.87	2.58%
1	40	119.94	116.91	2.59%
1	60	119.96	116.93	2.60%
1	80	119.97	116.93	2.60%
1	100	119.98	116.94	2.60%

controls the convergence of the values but does not significantly change the heights of the cantilever beam. The approximation errors come from the linear approximation of the optimal heights which are determined mainly by the design parameter A.

As we mentioned before, the upper and lower bounds in Equation (6.8) will affect the optimal solution of the design. Figure 6.5 shows the range of the optimal solution of the first natural frequency from 102 Hz to 120 Hz, which is the tolerance range of the desired upper bound of the unwanted frequency band 110 Hz. If the upper bound and lower bound in Equation (6.8) are set to u = 130 Hz and l = 90 Hz respectively, as shown in Figure 6.6, the optimal solution will be in the range from 90 Hz to 102 Hz which is the tolerance range of the desired lower bound 100 Hz. Figure 6.7 (a) and (b) show the optimal shapes with A = 100, B = 1, and A = 1, B = 100 for the upper bound and the lower bound set to 120 Hz and 80 Hz respectively. Figure 6.7 (c) and (d) show the optimal shapes for the upper bound and lower bound set to 130 Hz and 90 Hz respectively.

Figure 6.7 shows different optimal cantilever beams for different DWPs. It is obvious that different DWP setups resulting in different shape changes. The designer can thus take advantage of the visual display of the cantilever beam to decide the designer-weighted optimal shape based on the practical requirements or the prior knowledge of the design problem.

6.3.2 Engine Connecting Rod Example

6.3.2.1 Problem Definition

The engine connecting rod in Section 4.5 is again used to demonstrate the interactive 3D shape design optimization process in the virtual environment. Since node N_{13} and element E_{45} contain the highest displacement and the highest vonMises stress respectively, the objective is to find a minimum change of the current design to reduce the displacement and stress at least 5% in these two areas by changing the width, length, and the contour on the shank section of

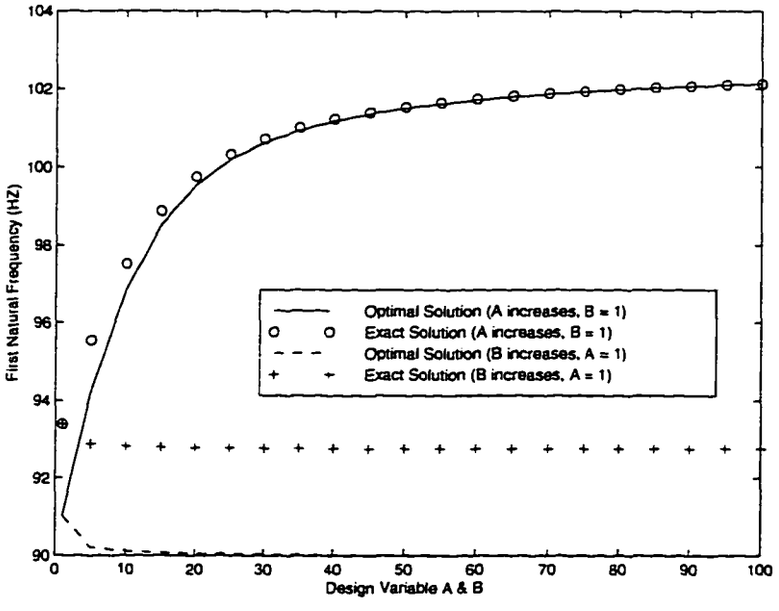
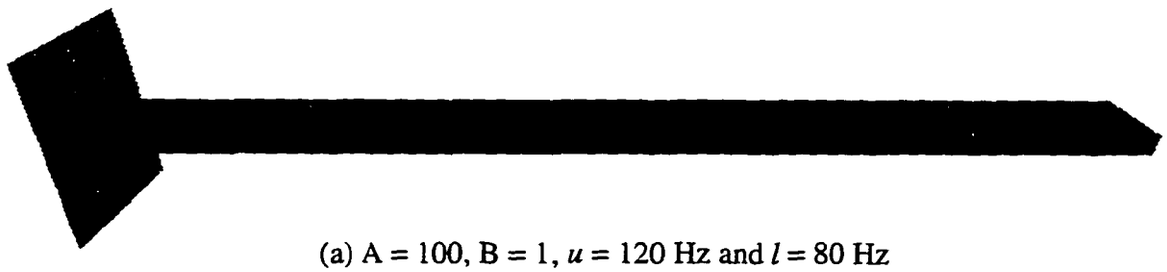


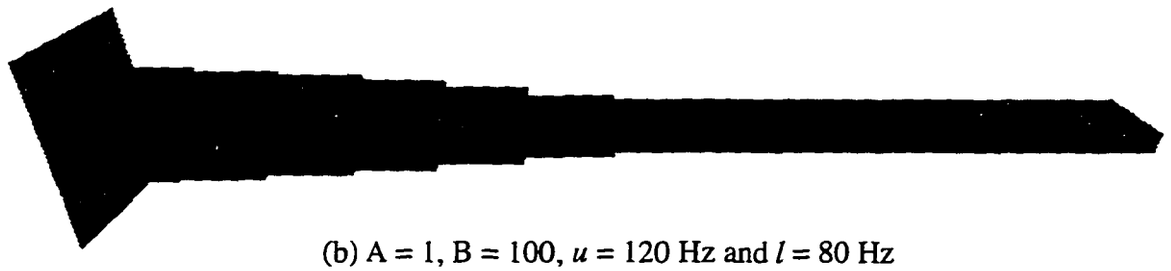
Figure 6.6: Comparison plot of the optimal solutions and the exact solutions for l = 90 Hz and u = 130 Hz

Original Landal = 102.71 HZ, Original Weight = 6.82 lb
Optimal Landal = 102.98 HZ, Optimal Weight = 6.84 lb



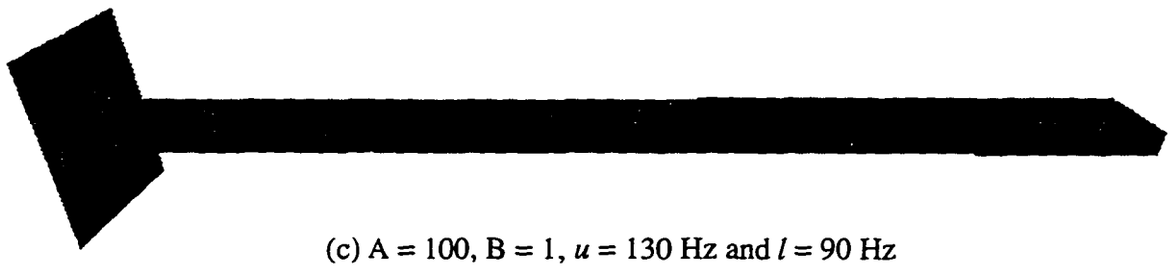
(a) A = 100, B = 1, u = 120 Hz and l = 80 Hz

Original Landal = 102.71 HZ, Original Weight = 6.82 lb
Optimal Landal = 119.98 HZ, Optimal Weight = 8.50 lb



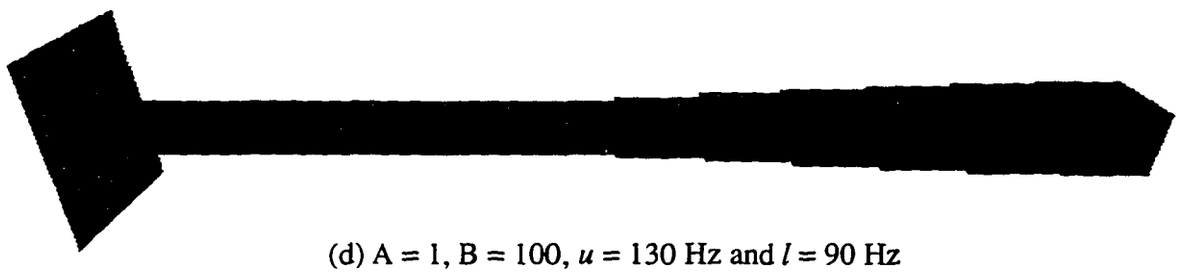
(b) A = 1, B = 100, u = 120 Hz and l = 80 Hz

Original Landal = 102.71 HZ, Original Weight = 6.82 lb
Optimal Landal = 102.13 HZ, Optimal Weight = 6.87 lb



(c) A = 100, B = 1, u = 130 Hz and l = 90 Hz

Original Landal = 102.71 HZ, Original Weight = 6.82 lb
Optimal Landal = 90.01 HZ, Optimal Weight = 7.89 lb



(d) A = 1, B = 100, u = 130 Hz and l = 90 Hz

Figure 6.7: Optimal solution of the cantilever beam with various design parameters

the connecting rod. The NURBS volume and the corresponding control point lattice should be properly defined in order to maintain the features of the pin end and crankshaft end while changing the design variable. In this case, the knot vectors are defined as follows:

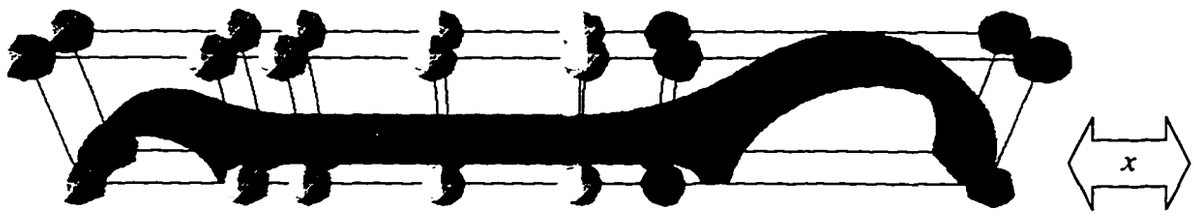
$$\mathbf{U} = \{0.0, 0.0, 0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.0, 1.0\}$$

$$\mathbf{V} = \{0.0, 0.0, 1.0, 1.0\}$$

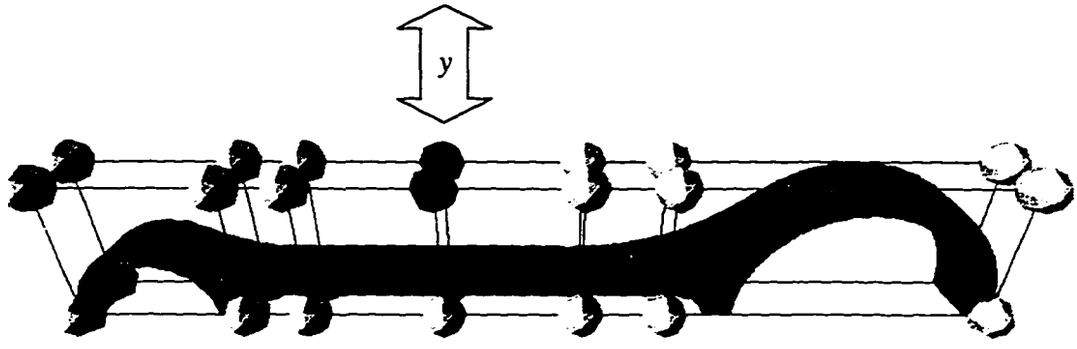
$$\mathbf{W} = \{0.0, 0.0, 1.0, 1.0\}$$

The layout of the control point lattice is shown in Figure 6.8. The design variables are shown as the highlighted control points in Figure 6.8 (a), (b) and (c) respectively.

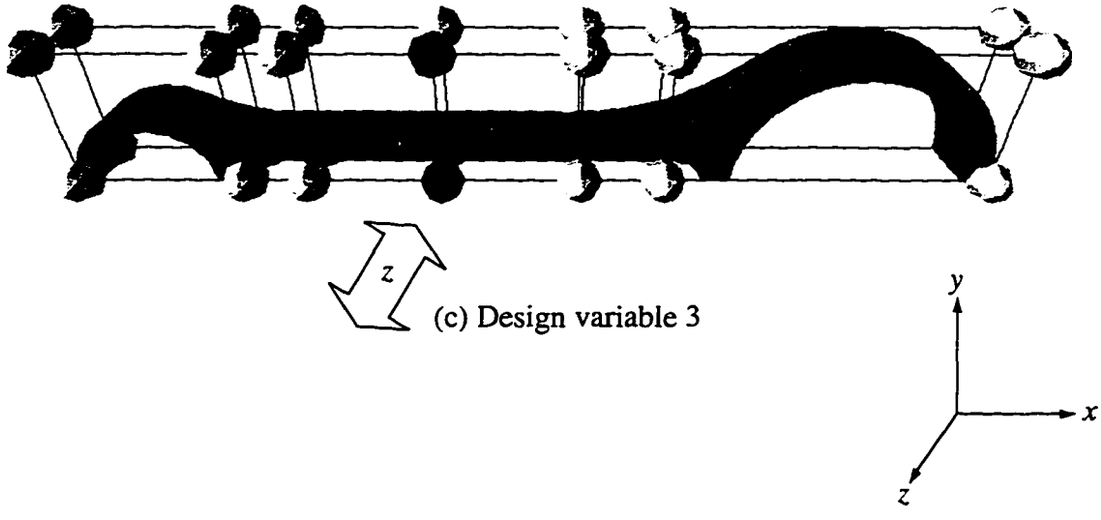
Note from Figure 6.8 that eight control points on the left and right of the control point lattice cover the whole area of the pin end and the crankshaft end respectively. The first design variable is the x direction movement of the eight control points on the right of the lattice, as shown in Figure 6.8 (a), to change the length of the connecting rod. By moving these control points simultaneously, the feature of the round end can be preserved. Also due to the local modification scheme, the movement of these control points will not affect the shape of the pin end. The second design variable is the y direction movement of the two control points as shown in Figure 6.8 (b). Similarly, the movement of these two control points will not affect the features of the two round ends. Since the knot vector $\mathbf{V}$ indicates the linear change in y direction, the interior finite element nodes will be changed linearly with the design variable change in order to maintain the regularity of the overall finite element mesh. The third design variable is the z direction movement of the 2 front control points as shown in Figure 6.8 (c).



(a) Design variable 1



(b) Design variable 2



(c) Design variable 3

Figure 6.8: The layout of the control point lattice and the design variables

It will change the width of the shank section. Also the interior nodes will be changed linearly in z direction to maintain the mesh integrity.

6.3.2.2 Formulation of the Interactive Design Optimization

To find the minimum changes of the design variables with 5% reduction of the displacement and stress, the objective function for the interactive design optimization is formulated as follows:

$$\textbf{\textit{minimize}} \qquad C(\boldsymbol{h}, d, \sigma) = A \cdot f(\boldsymbol{h}) + B \cdot g_1(d) + C \cdot g_2(\sigma) \tag{6.10}$$

$$f(\boldsymbol{h}) = \sum_{i=1}^{3} f_i(h_i) = \sum_{i=1}^{3} 1 + \cos(2\pi(h_i - u)/(l - u)), \qquad (l_h < h_i < u_h)$$

$$\begin{aligned} g_1(d) &= 1 - \cos(2\pi(d - l_d)/(2 \times (l_d - u_d))), && (l_d < d < u_d) \\ g_1(d) &= 2 && d \geq u_d \\ g_1(d) &= 0 && d \leq l_d \end{aligned}$$

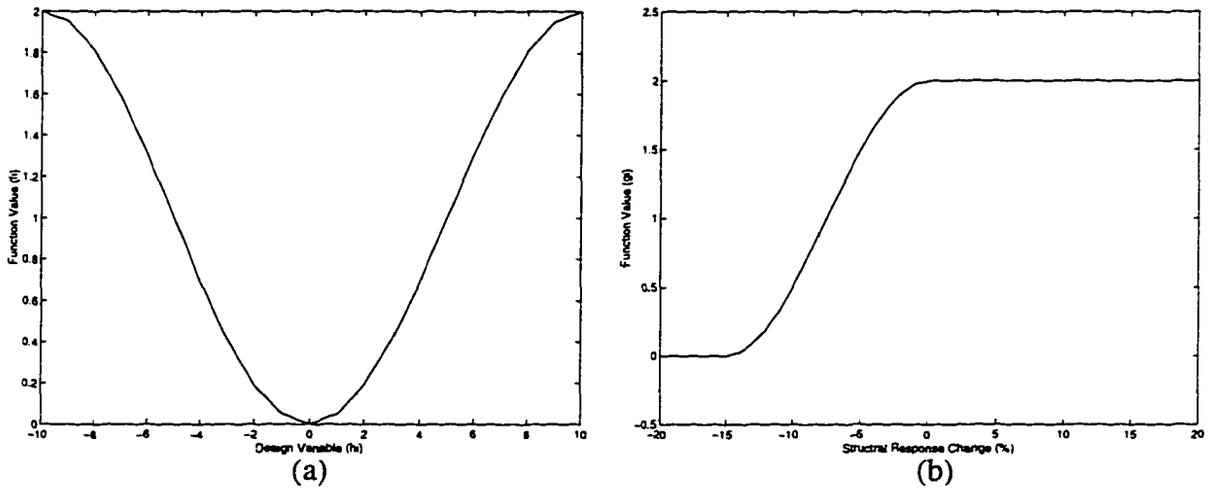
$$\begin{aligned} g_2(\sigma) &= 1 - \cos(2\pi(\sigma - u_\sigma)/(2 \times (l_\sigma - u_\sigma))), && (l_\sigma < \sigma < u_\sigma) \\ g_2(\sigma) &= 2 && \sigma \geq u_\sigma \\ g_2(\sigma) &= 0 && \sigma \leq l_\sigma \end{aligned}$$

where C is the cost function. A, B, and C are the DWPs. $\boldsymbol{h}$ is the vector of the design variables as illustrated in Figure 6.8. d and σ are the displacements and stresses obtained through linear approximation at node N_{13} and element E_{45} respectively. f is the function of the design variables which results in a higher value as the design variables approach their upper and lower

bounds. u_h and l_h are the upper bound and lower bound of the design variable, which are set to 10 and -10 in this case. g_1 and g_2 are the functions of the structural responses which produce a lower number as the displacement or stress is reduced. The upper bound u_d and u_σ, are the original values of the displacement and stress. The lower bound l_d and l_σ are set to 10% less than the original values of the displacement and stress in order to compensate the value of the function f such that the optimal solution of the displacement and stress can be less than 10% of the original values. Note that the function f for the design variable changes is not a quadratic function as in the cantilever beam example, but a trigonometric function with the value output range from 0 to 2 in order to normalize the cost function. The function plots of f_i and g are shown in Figure 6.9 (a) and (b) respectively.

6.3.2.3 Results

To perform the interactive design optimization, the sensitivities for all three design variables should be obtained first. The process to perform the sensitivity analysis in the virtual environment has been described in Section 5.4. After the sensitivity information is obtained,



(a) (b)

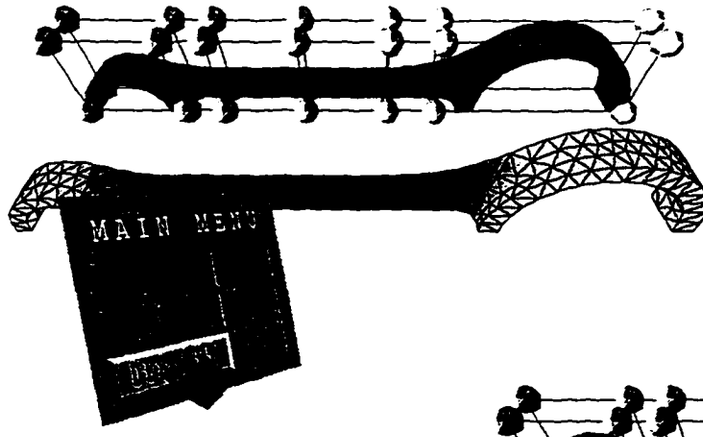
Figure 6.9: Function Plots of (a) f_i and (b) g

by pressing the button **SEN ANS**, the sensitivity data can be loaded into the program and the designer can start the interactive design optimization process by clicking the button **DES OPT** on the virtual menu (Figure 6.10 (a)). Another menu with three sliders will pop up as shown in Figure 6.10 (b). The designer can move the sliders to change the DWPs. After pressing the button **APPLY**, the updated shape and the corresponding analysis results as shown in Figure 6.10 (c) represent the designer-weighted optimal solution. Note that the optimal shape of the connecting rod is exaggerated by a scaling factor since the design variable changes are sometime too small to tell the difference from the visual feedback. The designer should be careful when interpreting the results from the virtual model.

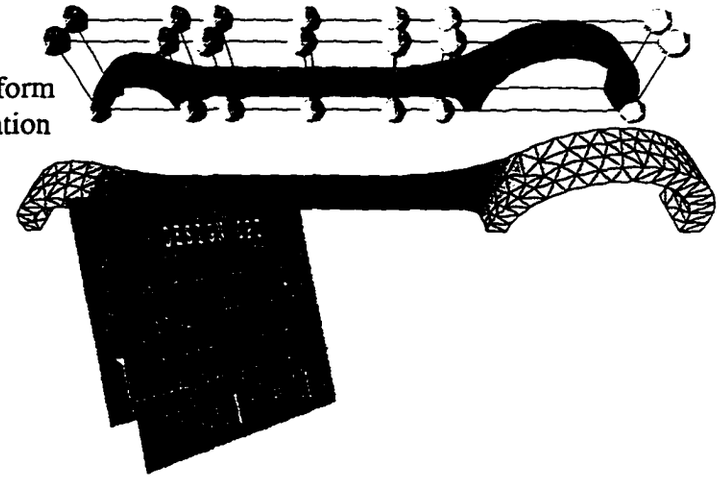
6.3.2.4 Design Verification

In the interactive design optimization process, the accuracy of the optimal solution is highly dependent on the accuracy of the sensitivity computation and the linearity of the structural responses with respect to the design variables. The sensitivity computation methods have been discussed in Chapter 5. In this example, the linear sensitivities for three design variables show reasonable accuracy to approximate the structural response of the modified shape. Figure 6.11 shows the comparison plots of the linear approximation and the finite element reanalysis for the three design variables.

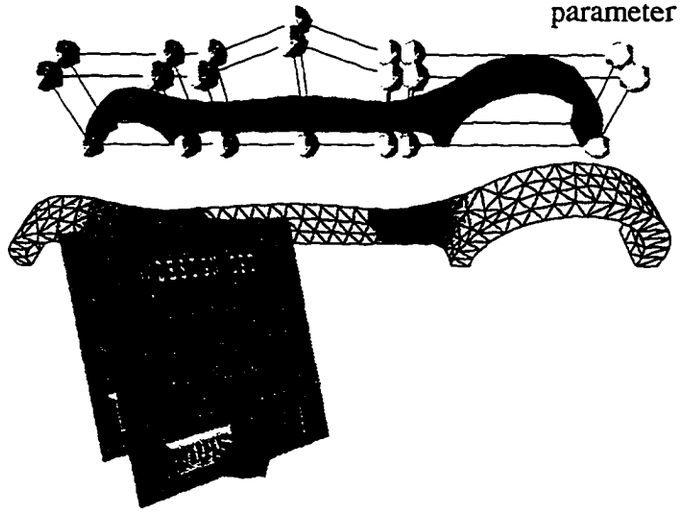
By using the interactive design optimization tool, the designer can easily perform trade-off analysis between the shape changes and the structural responses, and determine the best solution based on the prior knowledge or the practical requirements. For the engine connecting rod case, the displacement of node N_{13} and vonMises stress of element E_{45} are `5.858E-02` inches and `1.302E02` psi respectively, and the desired displacement and



(a) Press **DES OPT** button to perform the interactive design optimization task.



(b) Use PinchGlove to change the designer-weighted parameter by moving the sliders.



(c) Press **APPLY** to find the optimal solution of current designer-weighted parameter settings.

Figure 6.10: Interactive design optimization process in the virtual environment

stress are 5.565E-02 inches and 1.237E02 psi which are 5% less than the original values. However, these values are not rigid constraints but a design goal. The designer can easily perform the trade-off design by changing the DWPs to decide which design factor, overall design change or structural responses, is more important. Since the cost function is relatively easy to compute compared to the finite element reanalysis, the optimal solution can converge in a short time. The optimal shape is displayed and the optimal solution of the displacement and the stress are also shown on the screen. Tables 6.3, 6.4, and 6.5 show the comparison of the optimal solutions and the finite element reanalysis with various DWP settings. The designer can select any combination of the DWP settings and perform the design optimization until the optimal solution satisfies the design goal.

Table 6.3: Comparison of optimal solutions with finite element reanalysis by increasing Designer-Weighted Parameter 1 (DWP2 = 1, DWP3 = 1)

DWP 1	Optimal Solution (Displacement)	Finite Element Reanalysis (Displacement)	Error %	Optimal Solution (Stress)	Finite Element Reanalysis (Stress)	Error %
1	0.0535	0.0541	1.12%	130.393	130.387	0.004%
2	0.0542	0.0547	0.91%	130.358	130.354	0.003%
3	0.0550	0.0553	0.54%	130.326	130.324	0.002%
4	0.0558	0.0560	0.36%	130.294	130.295	0.00%
5	0.0567	0.0568	0.18%	130.259	130.260	0.00%
6	0.0586	0.0586	0.00%	130.198	130.198	0.00%
7	0.0586	0.0586	0.00%	130.198	130.198	0.00%
8	0.0586	0.0586	0.00%	130.198	130.198	0.00%
9	0.0586	0.0586	0.00%	130.198	130.198	0.00%
10	0.0586	0.0586	0.00%	130.198	130.198	0.00%

Table 6.4: Comparison of optimal solutions with finite element reanalysis by increasing Designer-Weighted Parameter 2 (DWP1 = 1, DWP3 = 1)

DWP 2	Optimal Solution (Displacement)	Finite Element Reanalysis (Displacement)	Error %	Optimal Solution (Stress)	Finite Element Reanalysis (Stress)	Error %
1	0.0535	0.0541	1.12%	130.393	130.387	0.004%
2	0.0531	0.0538	1.34%	130.414	130.386	0.005%
3	0.0530	0.0537	1.40%	130.421	130.404	0.008%
4	0.0529	0.0537	1.44%	130.425	130.411	0.008%
5	0.0529	0.0536	1.45%	130.427	130.415	0.008%
6	0.0528	0.0536	1.46%	130.429	130.416	0.008%
7	0.0528	0.0536	1.47%	130.430	130.418	0.008%
8	0.0528	0.0536	1.48%	130.431	130.419	0.009%
9	0.0528	0.0536	1.48%	130.431	130.420	0.009%
10	0.0528	0.0536	1.49%	130.432	130.420	0.009%

Table 6.5: Comparison of optimal solutions with finite element reanalysis by increasing Designer-Weighted Parameter 3 (DWP1 = 1, DWP2 = 1)

DWP 3	Optimal Solution (Displacement)	Finite Element Reanalysis (Displacement)	Error %	Optimal Solution (Stress)	Finite Element Reanalysis (Stress)	Error %
1	0.0535	0.0541	1.12%	130.393	130.387	0.004%
2	0.0535	0.0541	1.12%	130.393	130.387	0.004%
3	0.0535	0.0541	1.12%	130.393	130.387	0.004%
4	0.0535	0.0541	1.12%	130.393	130.387	0.004%
5	0.0535	0.0541	1.12%	130.393	130.387	0.004%
6	0.0535	0.0541	1.12%	130.393	130.387	0.004%
7	0.0535	0.0541	1.12%	130.393	130.387	0.004%
8	0.0535	0.0541	1.12%	130.393	130.387	0.004%
9	0.0535	0.0541	1.12%	130.393	130.387	0.004%
10	0.0535	0.0541	1.12%	130.393	130.387	0.004%

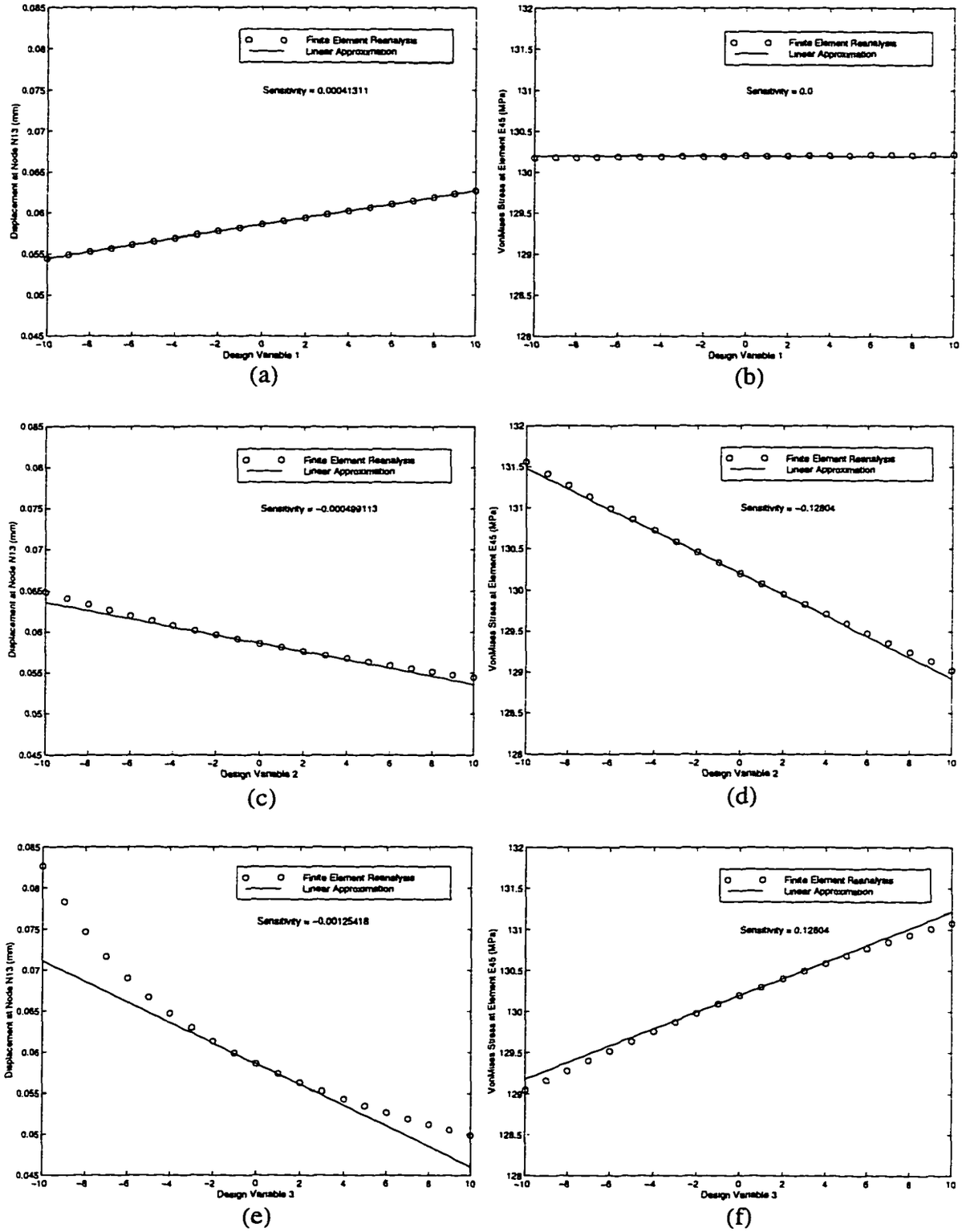


Figure 6.11: Comparison plots of linear approximation and finite element reanalysis for three design variables

Increasing the DWP1 indicates that the design is more conservative toward the overall shape changes. From the visual feedback of the virtual analysis result model as well as the numerical results listed in Table 6.3, the shape of the connecting rod tends to remain unchanged as DWP1 increases, so the displacement and the VonMises stress are also remained the same.

The DWP2 represents the designer's concern about the displacement at node N_{13}. Increasing DWP2 will decrease the displacement since reducing the displacement becomes more important in the cost function. Table 6.4 shows the same results as we expect. The percent error of the optimal solution for the displacement increases is due to the fact that the errors of the linear approximations for nonlinear models increase as the changes to the design variables become large.

Increasing DWP3 should decrease the vonMises stress at element E_{45} since the decreasing of the vonMises becomes more important in the cost function. However, a very interesting result comes from the analysis result model. The shape remains the same as we change the slider of DWP3. This could result from the fact that $g_2(\sigma)$ in the cost function Equation (6.10) remains the same value, i.e. $\sigma \geq u_\sigma$ or $\sigma \leq l_\sigma$, at the optimal result, thus the value of DWP3 does not affect the optimal solution. This may also indicate a bad selection of the design variables to cause $g_2(\sigma)$ to remain the same value.

By using such an interactive design optimization tool, it is easy to tell that the current selection of design variables need to be adjusted. However, further investigation is required to decide a better choice of the design variables.

From the sensitivity plots shown in Figure 6.11 (e) and (f), the sensitivities of design variable 3 for the displacement and vonMises stress are of opposite sign, which means that as design variable 3 decreases the displacement, the vonMises stress also increases. We can thus take out this design variable and perform the interactive design optimization again. Tables 6.6 and 6.7 show the comparison of the optimal solution with finite element reanalysis for only two design variables by increasing DWP2 and DWP3. As we expect, the displacement and vonMises stress decrease as DWP2 and DWP3 increase respectively.

6.4 Conclusion

The inclusion of interactive design optimization in the virtual environment is successfully implemented in this chapter. This method allows easy, quick design changes and evaluations to be performed in a 3D virtual design space. The concept of the cost function minimization is introduced. It is different from the traditional design optimization method in the following ways:

1. For the traditional design optimization, the design process starts from a preliminary design and the optimal shape is obtained after the optimization process has converged. The design problem definition will not be changed throughout the optimization process unless the optimal solution is not satisfied and a restart of the design is requested. However, using the interactive design optimization, the design process starts from a relatively good design and the designer explores different design changes by interactively modifying the design problem definition. This method allow the designer to investigate the trade-offs between conflicting constraints and seek the possibility of improvements for the current design. The optimal solution

Table 6.6: Comparison of optimal solutions with finite element reanalysis by increasing Designer-Weighted Parameter 2 (Two Design Variables)

DWP 2	Optimal Solution (Displacement)	Finite Element Reanalysis (Displacement)	Error %	Optimal Solution (Stress)	Finite Element Reanalysis (Stress)	Error %
1	0.0559	0.0561	0.19%	129.784	129.791	0.005%
2	0.0531	0.0540	1.59%	128.918	129.020	0.079%
3	0.0531	0.0539	1.59%	128.918	129.019	0.078%
4	0.0530	0.0539	1.60%	128.918	129.019	0.078%
5	0.0530	0.0538	1.60%	128.918	129.018	0.078%
6	0.0529	0.0538	1.60%	128.918	129.018	0.078%
7	0.0529	0.0538	1.60%	128.918	129.018	0.077%
8	0.0529	0.0538	1.60%	128.918	129.018	0.077%
9	0.0529	0.0537	1.60%	128.918	129.018	0.077%
10	0.0529	0.0537	1.60%	128.918	129.018	0.077%

Table 6.7: Comparison of optimal solutions with finite element reanalysis by increasing Designer-Weighted Parameter 3 (Two Design Variables)

DWP 3	Optimal Solution (Displacement)	Finite Element Reanalysis (Displacement)	Error %	Optimal Solution (Stress)	Finite Element Reanalysis (Stress)	Error %
1	0.0559	0.0561	0.19%	129.784	129.791	0.005%
2	0.0559	0.0560	0.20%	129.773	129.781	0.006%
3	0.0559	0.0560	0.21%	129.762	129.771	0.006%
4	0.0558	0.0560	0.22%	129.750	129.759	0.007%
5	0.0558	0.0559	0.23%	129.737	129.747	0.008%
6	0.0557	0.0558	0.24%	129.723	129.734	0.009%
7	0.0556	0.0558	0.26%	129.708	129.720	0.009%
8	0.0556	0.0557	0.28%	129.691	129.704	0.010%
9	0.0556	0.0557	0.30%	129.672	129.686	0.011%
10	0.0555	0.0556	0.32%	129.649	129.665	0.012%

of the interactive design optimization is dependent on the designer's knowledge of the design problem or the practical requirement to adjust the DWPs.

2. The objective function and constraints in the traditional design optimization method often take considerable time to evaluate in every design iteration. Since the updated structural responses should be calculated in every small design variable change, it is a time consuming process if the design model is complicated. In the interactive design optimization, the structural responses in every design step are obtained through sensitivity approximations. The cost function in Equation (6.8) is also simplified as a quadratic or trigonometric function. Thus for a small number of design variables, the optimization solution for the interactive design process can converge in almost real time.

3. In the traditional design optimization formulation, the design objective and the design constraints should be defined explicitly thus a thorough understanding of the design problem is required. If the optimal solution is not satisfied, the designer has to modify the design problem definition and restart the design optimization. In the interactive design optimization, the problem definition can be interactively changed through the DWPs. Since the purpose of the interactive design optimization is to find the most valuable design within different constraint settings, the design constraints are not set up rigidly but can be changed in the design process. Through the fast approximation method, the designer can quickly determine the optimal solution of different problems thus giving the designer an intuitive sense of what effect the design changes have on the optimal solution, and helps the designer to decide on the trade-offs in a timely manner.

4. As shown in the engine connecting rod example, the interactive design optimization can be used to make the decision on the choice of the design variables. As discussed in Chap-

ter 5, the sensitivity analysis makes it easy to relate the structural responses to the changes of the design variables. By playing with the DWPs in the interactive design optimization process, the relationship between the design variables and the performance of the optimal shape becomes more intuitive to the designer and it will help the designer to decide on a proper set of the design variables. Although this tool is not as accurate of a design tool as the traditional design optimization process, the efficiency of the interactive design optimization makes it an excellent choice for the design refinement stage and for the real-time interaction in a virtual environment.

CHAPTER 7. CONCLUSIONS AND FUTURE WORK

This research has successfully incorporated virtual reality techniques, NURBS-based free-form deformation, sensitivity analysis, and designer-weighted optimization to facilitate interactive structural shape design. NURBS-based free form deformation has been combined with displacement and stress sensitivities to allow a designer to change the shape of a design and immediately view the deformation and stress distribution changes in a virtual environment. Designer-weighted optimization is implemented to guide the designer to investigate the designs in the vicinity of an optimal design. The virtual environment implementation allows the designer to select design variables, change the shape of the design, and view the resulting changes using natural three-dimensional movements. Stereo viewing further enhances the ability of the designer in concentrating on the evaluation of the analysis results and provides a natural interface to interact with computer images and databases. Several examples have demonstrated the advantages of using VR as a design tool for structural shape design optimization.

Current limitations of the VR program developed in this research have been discussed in Chapters 3, 5, and 6. The limitations are summarized as following and are provided as suggestions for future improvements:

1. This program is designed to modify the structural shape for static and normal mode analysis only, and the acceptable finite element types are limited as listed in Table 3.1. The limitation is based on the file translation from MSC/NASTRAN data files to the database of this program.
2. The visualization capability of this program can be improved through combination with the CAVE libraries such that this program can be executed in the C2 as a fully

immersive virtual environment. The interaction capability can also be improved by incorporating the voice recognition system for vocal command inputs.

3. The design model should be relatively rectangular in shape in order to maintain the regularity of the finite element mesh after deformation, due to the use of a rectangular bounding box to define the NURBS surface or volume.
4. The shape of the design model where the boundary condition is applied can not be modified in order to obtain an accurate linear sensitivity of the displacement and stress.
5. Movements of the design variables are limited in a small range in this application due to the use of linear sensitivity analysis. Higher order approximations can be used to extend the sensitivity method for large design changes.

As was discussed in Chapter 2, VR has been applied in many fields including design and manufacturing. However, an integrated virtual environment to perform concurrent engineering tasks from design to marketing is still lacking. This research has shown promise by developing an approach to integrate structural shape modification, analysis results, sensitivity information, and design optimization techniques in a virtual environment. The ultimate goal of this approach is to build a virtual environment for concurrent engineering including design, analysis, manufacturing, and marketing. Figure 7.1 shows the overall structure of the virtual environment and its applications for concurrent engineering. The highlighted blocks indicate the current achievements of this research in the role of this integrated virtual environment.

The primary challenge of this integrated VR application is to build a common data base which contains the geometry information, analysis results, assembly hierarchy, and material properties for rendering, etc. The common data base should be general enough to support the

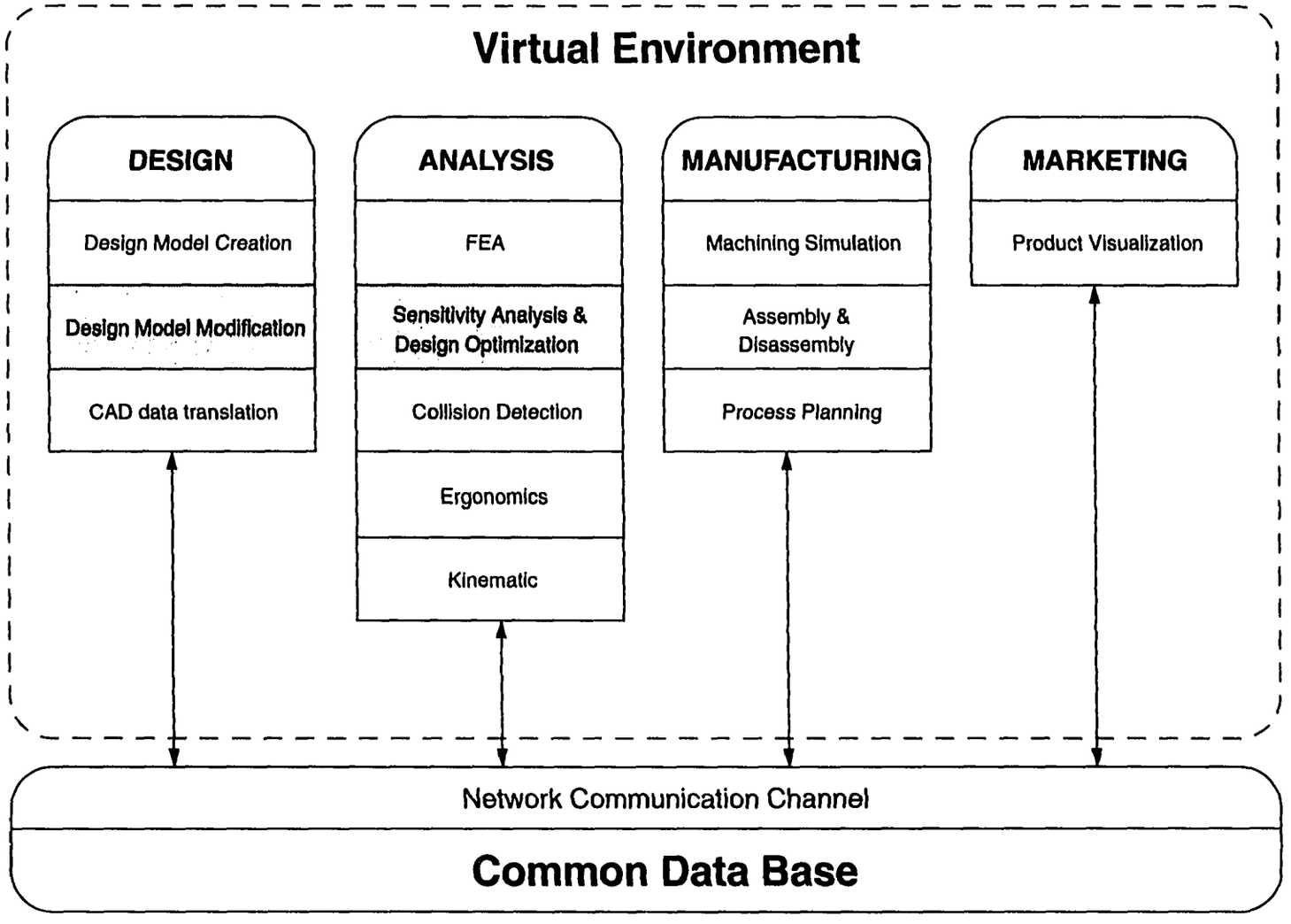


Figure 7.1: The integrated virtual environment for concurrent engineering

required information for different tasks. It should also have extensibility to accommodate new tasks into this environment. The network communication channel is a mechanism to connect the applications with proper data in the common data base, and it also contains the capability to transfer the data file over a low bandwidth network connection, for example the Virtual Reality Modeling Language (VRML) file format [109], such that the common database can be shared with remote designers or even marketing representatives over the network. Due to the intuitive and immersive nature of VR, the design concept can be easily illustrated and interpreted with a digital mock-up in a virtual environment. With this kind of integrated virtual environment, all the design team members with different areas of expertise, from design to marketing, can contribute their knowledge to bear on solving the problem of the design model based on a common data base, which will result in an innovative and competitive product designed and produced in a shorter time frame.

APPENDIX A. CONFIGURATION FILE FORMAT FOR THE VIRTUAL ENVIRONMENT INITIALIZATION

The configuration file serves to provide the parameter specifications to initialize the virtual environment. It contains all the necessary data files to be loaded into the virtual environment and the initial set up of the virtual design object and its corresponding analysis results. The standard format for each statement in the configuration file is as follows:

configuration_parameter <parameter value entry>

A sample configuration file is listed in section A.1, and the reference guide for each statement is listed in section A.2 in alphabetical order. Note that the configuration file can be extended to meet the designer's needs, and the program for building the virtual environment should be modified to accommodate the new specifications.

A.1 Sample Configuration File

```
# Sample configuration file

# Data file input
nastran_dat filename.dat
nastran_pch filename.pch
nastran_sen_f06 filename.f06
nastran_sen_dscm filename.dscm2
nastran_sen_dvf filename.desvcp

# Color index
num_color_id 10

# RGB color of the Color bar
color_id 0 0 120
color_id 0 0 240
color_id 0 120 240
color_id 0 240 240
color_id 0 240 0
```

```
color_id 240 240 0
color_id 240 120 0
color_id 240 60 0
color_id 250 0 0
color_id 180 0 0

# Color of the model
color_md 255 191 193
color_hl 200 150 150

# Analysis result specifications

ans_exfactor 5.0
ans_loadcase 1

# Knot vector specification
num_knot 6 4 2
knot_u 0.0 0.0 0.0 1.0 1.0 1.0
knot_v 0.0 0.0 1.0 1.0
knot_w 0.0 1.0

# Control point latices specifications
num_cpt 3 2 1
cpt_u 0.0 0.5 1.0
cpt_v auto
cpt_w auto

# Virtual environment specifications
dist_fem_sen 0.0 -30.0 0.0
scale_sensor 6.0
scale_manip 2.0
design_variable_type 0

# end
```

A.2 Reference Guide for the Configuration File

Configuration Parameter	Parameter Value Entry
#	Comment line remark. Used to insert comments into the input file. Comment statements may appear anywhere within the configuration file.
ans_exfactor	Analysis results exaggeration factor. Used to multiply the analysis results by this factor. Default value is 10.0.
ans_loadcase	Load case number. Used to specify the load case of the analysis results in the virtual environment initially. Default value is 1.
color_id	The RGB color index for the stress contour. The number of this specification appearing in the configuration file should be the same as the number of **num_color_id.**
color_hl	The highlighted RGB color index. Used to specify the color of the design model after the parametrization process is completed.
color_md	The RGB color index of the design model.
cpt_u	Control Point Lattice in the U direction. Used to specify the initial control point position. If `auto` is specified, the control point lattice in the U direction is evenly divided.
cpt_v	Control Point Lattice in the V direction. Used to specify the initial control point position. If `auto` is specified, the control point lattice in the V direction is evenly divided.
cpt_w	Control Point Lattice in the W direction. Used to specify the initial control point position. If `auto` is specified, the control point lattice in the W direction is evenly divided.
design_variable_type	Design variable type. If 0, the control point is defined as the design variable for indirect manipulation. If 1, the finite element node is defined as the design variable for direct manipulation. Default value is 0.

Configuration Parameter	Parameter Value Entry
dist_fem_sen	Initial distance between the design model and the analysis result model. Default value is 0.0, 0.0, 0.0.
knot_u	Knot vector in the U direction.
knot_v	Knot vector in the V direction.
knot_w	Knot vector in the W direction.
nastran_dat	MSC/NASTRAN input data file name. Used to draw the design model in the virtual environment.
nastran_sen_dscm	MSC/NASTRAN DSCM file name. Contains the DSCM data block to specify the sensitivities of the displacement and the stress.
nastran_sen_dvf	MSC/NASTRAN DVF file name. Contains the information of the design velocity field.
nastran_sen_f06	MSC/NASTRAN f06 file name. Specifies the file format of the DSCM file.
nastran_pch	MSC/NASTRAN punch file name. Used to draw the analysis result model in the virtual environment.
num_color_id	Number of color indices. Used to specify the number of color contour. Default value is 10.
num_cpt	Number of control points in three parametric directions.
num_knot	Number of knot vector entries for **knot_u**, **knot_v**, and **knot_w**.
scale_sensor	Scaling factor of the sensor sensitivity.
scale_manip	Scaling factor of the size of the manipulator (the virtual hand model).

APPENDIX B. SAMPLE MSC/NASTRAN INPUT FILE FOR THE SENSITIVITY ANALYSIS

As was discussed in Chapter 3, a MSC/NASTRAN input file for the sensitivity analysis will be generated after the design variable is specified. The input file is modified from the original file used to perform the finite element analysis and draw the design model in the virtual environment. In this section, the input file and the corresponding specifications will be briefly introduced. A sample MSC/NASTRAN input file for the sensitivity analysis is presented in Section B.1. The highlighted blocks indicate the information automatically appended to the original input file by the program. The content of the highlighted blocks will be specified in Section B.2. Detail information can also be found in references [45][110].

B.1 Sample Input Data File

B.1.1 Determining Structural Response Sensitivities

```
assign output4='_output.dscm2',unit=11,delete,formatted
SOL 200  $ sensitivity analysis                                    (B1)
  TIME    =     60
compile exitopt nolist noref
alter 30 $
output4 dscm2,,,,//-2/iunit/-1//15 $                               (B2)
endalter
CEND
TITLE     = NASTRAN FILE TRANSLATOR  -- UNITS = MM
analysis = statics
desobj(min)  = 1                                                   (B3)
dessub = 1
$ GLOBAL CASE
  SPC     =      1
  LOAD    =     10
  DISPLACEMENT(PUNCH,REAL) = ALL
  STRESS(PUNCH,REAL) = ALL
BEGIN BULK
param,optexit,-4
param,nasprt,1                                                     (B4)
param,despch,1
```

```
PARAM   AUTOSPC       YES
PARAM   POST           -2
GRID            1       0 25.0000 7.50000 0.00000       0
GRID            2       0-25.0000 7.50000 0.00000       0
GRID            3       0 25.0000-7.50000 0.00000       0
GRID            4       0-25.0000-7.50000 0.00000       0
GRID            5       0 6.30E-4 3.00000 0.00000       0
GRID            6       0 3.00000 0.00000 0.00000       0
GRID            7       0-3.00000 -1.E-16 0.00000       0
GRID            8       0 6.30E-4-3.00000 0.00000       0
GRID            9       0 22.5000 7.50000 0.00000       0
GRID           10       0 20.0000 7.50000 0.00000       0
.............
CQUAD4          1       1      79      76      69      70 0.00000
CQUAD4          2       1      78      77      76      79 0.00000
CQUAD4          3       1      16      15      77      78 0.00000
CQUAD4          4       1      57      79      70       6 0.00000
CQUAD4          5       1      71      78      79      57 0.00000
CQUAD4          6       1      17      16      78      71 0.00000
CQUAD4          7       1      76      74      68      69 0.00000
CQUAD4          8       1      77      75      74      76 0.00000
CQUAD4          9       1      15      14      75      77 0.00000
CQUAD4         10       1      74      80      67      68 0.00000
.............
MAT1            1 3.00E+7          0.30000 7.32E-4
PSHELL          1       1 1.00000       1               1         1.0E-41
SPC1            1  123456      52THRU          56
FORCE           1       1         10000.0     0.0    -1.0      0.0
LOAD           10 1.00000 1.00000       1

$-------2-------3-------4-------5-------6-------7-------8-------9-------0
$           Analysis Model
$-------2-------3-------4-------5-------6-------7-------8-------9-------0
desvar  1       desv_x  1.0     -5.0    5.0
desvar  2       desv_y  1.0     -5.0    5.0                                  (B5)
desvar  3       desv_z  1.0     -5.0    5.0
```

```
dresp1  1       weight  weight
dresp1  2       disp    disp                    123             1       +   (B6)
+       2       3       4       5       6       7       8       9       +
+       10      11      12      13      14      15      16      17      +
+       18      19      20      21      22      23      24      25      +
+       26      27      28      29      30      31      32      33      +
+       34      35      36      37      38      39      40      41      +
+       42      43      44      45      46      47      48      49      +
+       50      51      52      53      54      55      56      57      +
+       58      59      60      61      62      63      64      65      +
+       66      67      68      69      70      71      72      73      +
+       74      75      76      77      78      79      80      81      +
+       82      83      84      85      86      87      88      89      +
+       90      91      92      93      94      95      96      97      +
+       98      99      100     101     102     103     104     105     +
+       106     107     108     109     110     111     112     113     +
+       114     115     116     117     118     119     120     121     +
+       122     123     124     125     126     127     128     129     +
+       130     131     132     133     134     135     136     137     +
+       138     139     140     141     142     143     144     145     +
```

```
+       146
dresp1  3       stress  stress  PSHELL          9               1
```
(B6)

```
dconstr 1       2       -1.0    1.0
dconstr 1       3       -1.0    1.0
```
(B7)

```
dscreen disp    -1.0E10 438
dscreen stress  -1.0E10 116
```
(B8)

```
dvgrid  1       9       0       0.00530 1.0     0.0     0.0
dvgrid  1       10      0       0.02235 1.0     0.0     0.0
dvgrid  1       11      0       0.05344 1.0     0.0     0.0
dvgrid  1       12      0       0.10170 1.0     0.0     0.0
dvgrid  1       13      0       0.17164 1.0     0.0     0.0
dvgrid  1       14      0       0.27020 1.0     0.0     0.0
dvgrid  1       15      0       0.40905 1.0     0.0     0.0
dvgrid  1       16      0       0.58975 1.0     0.0     0.0
dvgrid  1       17      0       0.70958 1.0     0.0     0.0
dvgrid  1       18      0       0.74952 1.0     0.0     0.0
dvgrid  1       19      0       0.70957 1.0     0.0     0.0
dvgrid  1       20      0       0.58974 1.0     0.0     0.0
dvgrid  1       21      0       0.40904 1.0     0.0     0.0
dvgrid  1       22      0       0.27018 1.0     0.0     0.0
dvgrid  1       23      0       0.17162 1.0     0.0     0.0
dvgrid  1       24      0       0.10169 1.0     0.0     0.0
dvgrid  1       25      0       0.05343 1.0     0.0     0.0
dvgrid  1       26      0       0.02235 1.0     0.0     0.0
dvgrid  1       27      0       0.00530 1.0     0.0     0.0
dvgrid  2       9       0       0.00530 0.0     1.0     0.0
dvgrid  2       10      0       0.02235 0.0     1.0     0.0
dvgrid  2       11      0       0.05344 0.0     1.0     0.0
dvgrid  2       12      0       0.10170 0.0     1.0     0.0
dvgrid  2       13      0       0.17164 0.0     1.0     0.0
dvgrid  2       14      0       0.27020 0.0     1.0     0.0
dvgrid  2       15      0       0.40905 0.0     1.0     0.0
dvgrid  2       16      0       0.58975 0.0     1.0     0.0
dvgrid  2       17      0       0.70958 0.0     1.0     0.0
dvgrid  2       18      0       0.74952 0.0     1.0     0.0
dvgrid  2       19      0       0.70957 0.0     1.0     0.0
dvgrid  2       20      0       0.58974 0.0     1.0     0.0
dvgrid  2       21      0       0.40904 0.0     1.0     0.0
dvgrid  2       22      0       0.27018 0.0     1.0     0.0
dvgrid  2       23      0       0.17162 0.0     1.0     0.0
dvgrid  2       24      0       0.10169 0.0     1.0     0.0
dvgrid  2       25      0       0.05343 0.0     1.0     0.0
dvgrid  2       26      0       0.02235 0.0     1.0     0.0
dvgrid  2       27      0       0.00530 0.0     1.0     0.0
dvgrid  3       9       0       0.0     0.0     0.0     1.0
dvgrid  3       10      0       0.0     0.0     0.0     1.0
dvgrid  3       11      0       0.0     0.0     0.0     1.0
dvgrid  3       12      0       0.0     0.0     0.0     1.0
dvgrid  3       13      0       0.0     0.0     0.0     1.0
dvgrid  3       14      0       0.0     0.0     0.0     1.0
dvgrid  3       15      0       0.0     0.0     0.0     1.0
dvgrid  3       16      0       0.0     0.0     0.0     1.0
dvgrid  3       17      0       0.0     0.0     0.0     1.0
dvgrid  3       18      0       0.0     0.0     0.0     1.0
```
(B9)

```
dvgrid  3       19      0       0.0     0.0     0.0     1.0
dvgrid  3       20      0       0.0     0.0     0.0     1.0
dvgrid  3       21      0       0.0     0.0     0.0     1.0
dvgrid  3       22      0       0.0     0.0     0.0     1.0
dvgrid  3       23      0       0.0     0.0     0.0     1.0
dvgrid  3       24      0       0.0     0.0     0.0     1.0
dvgrid  3       25      0       0.0     0.0     0.0     1.0
dvgrid  3       26      0       0.0     0.0     0.0     1.0
dvgrid  3       27      0       0.0     0.0     0.0     1.0
```

(B9)

```
bndgrid 123456  1       2       3       4       5       6       7       +
+       8       9       10      11      12      13      14      15      +
+       16      17      18      19      20      21      22      23      +
+       24      25      26      27      28      29      30      31      +
+       32      33      34      35      36      37      38      39      +
+       40      41      42      43      44      45      46      47      +
+       48      49      50      51      52      53      54      55      +
+       56      57      58      59      60
ENDDATA
```

(B10)

B.1.2 Determining Design Velocity Field

```
assign output4='output.desvcp',unit=11,delete,formatted
SOL 200  $ sensitivity analysis
  TIME    =     60
```

(B11)

```
compile desopt nolist noref souin=mscsou
alter 'DOPR2'
output4 desvcp,,,,//-2/11/-1//15 $
matprn desvcp // $ print output
endalter
CEND
TITLE     = NASTRAN FILE TRANSLATOR  -- UNITS = MM
```

(B12)

```
analysis = statics
desobj(min)  = 1
dessub = 1
$ GLOBAL CASE
  SPC     =      1
  LOAD    =     10
  DISPLACEMENT(PRINT,REAL) = ALL
  STRESS(PRINT,REAL) = ALL
BEGIN BULK
```

(B13)

```
$ param,optexit,-4
param,nasprt,1
param,despch,1
PARAM    AUTOSPC       YES
PARAM    POST           -2
GRID            1       0 25.0000 7.50000 0.00000       0
GRID            2       0-25.0000 7.50000 0.00000       0
GRID            3       0 25.0000-7.50000 0.00000       0
GRID            4       0-25.0000-7.50000 0.00000       0
GRID            5       0 6.30E-4 3.00000 0.00000       0
```

(B14)

```
GRID            6       0 3.00000 0.00000 0.00000       0
GRID            7       0-3.00000 -1.E-16 0.00000       0
GRID            8       0 6.30E-4-3.00000 0.00000       0
GRID            9       0 22.5000 7.50000 0.00000       0
GRID           10       0 20.0000 7.50000 0.00000       0
.................
CQUAD4          1       1      79      76      69      70 0.00000
CQUAD4          2       1      78      77      76      79 0.00000
CQUAD4          3       1      16      15      77      78 0.00000
CQUAD4          4       1      57      79      70       6 0.00000
CQUAD4          5       1      71      78      79      57 0.00000
CQUAD4          6       1      17      16      78      71 0.00000
CQUAD4          7       1      76      74      68      69 0.00000
CQUAD4          8       1      77      75      74      76 0.00000
CQUAD4          9       1      15      14      75      77 0.00000
CQUAD4         10       1      74      80      67      68 0.00000
.................
MAT1            1 3.00E+7          0.30000 7.32E-4
PSHELL          1       1 1.00000       1               1         1.0E-41
SPC1            1  123456      52THRU          56
FORCE           1       1         10000.0     0.0    -1.0     0.0
LOAD           10 1.00000 1.00000       1

$-------2-------3-------4-------5-------6-------7-------8-------9-------0
$           Analysis Model
$-------2-------3-------4-------5-------6-------7-------8-------9-------0
.................
$ this part is the same as the file in Section B.1.2
.................
ENDDATA
```

B.2 Input Data File Specification for the Sensitivity Analysis

Input Field	Descriptions
B1	Specify the output file name of the DSCM2 data block to be `_output.dscm2`, and change the original solution sequence to `SOL 200`.
B2	Insert the DMAP alter to output the DSCM2 data block.
B3	Define the analysis type and specify the objective function to be minimized.
B4	Define the parameters, where `OPTEXIT -4` is used to output the sensitivity results into the file specified in B1.

Input Field	Descriptions
B5	Define the design variables and their upper and lower bounds. Since any design variable specified in the virtual environment can have 3 dimensional movements, it is considered as three independent design variables in this file.
B6	Specify the design responses to be weight, displacement, and stress.
B7	Specify the upper and lower bounds of the design response for the objective function. Only used in the optimization process.
B8	Specify the constraints of the structural response. The truncation threshold is set on all possible nodes and elements thus the sensitivity information on all nodes and elements can be output to the file.
B9	Specify the design velocity field of three design variables for the movements in three Cartesian coordinates.
B10	Specify the boundary grid.
B11	Same as B1, the output file for the DESVCP data block is set to `output.desvcp`.
B12	Insert the DMAP alter to output the DESVCP data block.
B13	Same as B3.
B14	Same as B4. The parameter `OPTEXIT` should not be set in this case.

BIBLIOGRAPHY

[1] Aukstakalnis, S and D. Blatner, *Silicon Mirage: The Art and Science of Virtual Reality*, Peachpit Press, Berkeley. 1992.

[2] Vance, J. M., "Virtual Reality: What Potential Does It Hold for Engineering?", *Current Advances in Mechanical Design and Production, Sixth Cairo University International MDP Conference*, Cairo, Elsevier Science, New York, N.Y., pp. 333-348, January 2-4, 1996.

[3] Gupta, R., "Survey on Use of Virtual Environments in Design and Manufacturing," *Proceedings of the 1996 ASME Design Engineering Technical Conference and Computers in Engineering Conference*, Irvine, CA. August 18-22, 1996.

[4] Rheingold, H., *Virtual Reality*, Simon & Schuster, New York, NY, 1991.

[5] Heilig, M., US Patent #3,050,870, *Sensorama Simulator*, August 28, 1962.

[6] Heilig, M., US Patent #2,955,156, *Stereoscopic-Television Apparatus for Individual Use*, October 4, 1960.

[7] Sutherland, I. E., "The Ultimate Display," *Proceedings of IFIP Congress*, pp. 506-508, 582-583, 1965.

[8] Sutherland, I. E., "A Head-Mounted Three Dimensional Display," *Proceedings of the Fall Joint Computer Conference*, Vol. 33, pp. 757 - 764, 1968.

[9] Krueger, M. *Artificial Reality II.* Addison and Wesley, Reading, MA, 1991.

[10] Bolt, R. "Put-That-There: Voice and Gesture at the Graphic Interface." *Proceedings of SIGGRAPH' 80*, Vol. 14, No. 3, pp. 262-270, July 1980.

[11] Burdea, G. and P. Coiffet, *Virtual Reality Technology*, John Wiley & Sons Inc. New York, NY, 1994.

[12] Adam, J. A., "Virtual Reality is for Real," *IEEE Spectrum*, Vol. 30, No. 10, pp. 22-29, October 1993.

[13] Azuma, R., "A Survey of Augmented Reality," *Course Notes #9, Developing Advanced Virtual Reality Applications, ACM SIGGRAPH*, Los Angeles, CA, pp 20.1-20.38, August 6-11, 1995.

[14] Gottschalk, M. A., "Engineering Enters the Virtual World: VR technology steps out of the arcade and into the engineering tool box," *Design News*, Vol. 49 <50>, pp. 23-24, May 9, 1994.

[15] Rosenblum, L. J., "Distributed Virtual Reality: Supporting Remote Collaboration in Vehicle Design," *IEEE Computer Graphics and Applications*, Vol. 17, No. 2, pp 13-17, March-April, 1997.

[16] Deitz, D., "Real Engineering in a Virtual World," *Mechanical Engineering*, Vol. 117, pp. 78-85, July 1995.

[17] Mahoney, P. D., "Driving VR," *Computer Graphics World*, pp. 22-33, May 1995.

[18] Owen, J. V., "Making Virtual Manufacturing Real," *Manufacturing Engineering*, Vol. 113, No. 5, pp. 33-37, November 1994.

[19] Schmitz, B. "Virtual Reality: On the Brick of Greatness," *Computer-Aided Engineering*, Vol. 12, No. 4, pp. 26-32, 1993.

[20] Cruz-Neira, C., Sandin, D. J., DeFanti, T. A., Kenyon, R. V. and J. C. Hart, "The CAVE: Audio-Visual Experience Automatic Virtual Environment," *Communication of the ACM*, Vol. 35, No. 6, pp. 65-72, 1992.

[21] Cruz-Neira, C., "Virtual Reality Overview," *ACM SIGGRAPH' 93 Course Notes: Applied Virtual Reality, ACM SIGGRAPH' 93 Conference*, Anaheim, California, pp. 1.1 - 1.18, August 1-6, 1993.

[22] *Virtual Reality: Scientific and Technological Challenges*, National Academy Press, Washington, D.C., 1995.

[23] Coull, T., "Mission Accomplished," *NASA Tech. Briefs*, pp. 16-17, July 1993.

[24] Wickens, C. D. and P. Baker, "Cognitive Issues in Virtual Reality," *Virtual Environments and Advanced Interface Design*, Barfield, W. and T. A. Furness (Eds.), Oxford University Press, New York, NY, 1995.

[25] Ellis, G., "Digital Clay: Transforming Automobile Design," *IRIS Universe*, No. 37, pp. 28-32, 1996.

[26] Beier, K. P., "Virtual Reality in Automotive Design and Manufacturing," *Proceedings of Convergence*, SAE Congress, pp. 241-247, October 1994.

[27] Dani, T. H. and R. Gadh, "COVIRDS: A Conceptual Virtual Design System," *Proceedings of the 15th ASME International Computers in Engineering Conference*, Boston, MA, September 17-21, 1995.

[28] Dani, T. H. and R. Gadh, "Creation of Concept Shape Designs via a Virtual Reality Interface," *Computer-Aided Design*, Vol. 29, No. 8, pp. 555-563, August 1997.

[29] Trika, S. N., Banerjee, P. and R. L. Kashyap, "Virtual Reality Interfaces for Feature-Based Computer-Aided Design Systems," *Computer-Aided Design*, Vol. 29, No. 8, pp. 565-574, August 1997.

[30] Osborn, S. W. and J. M. Vance, "A Virtual Reality Environment for Synthesizing Spherical Four-Bar Mechanisms," *Proceedings of the ASME Design Automation Conference*, Boston, MA, September 17-21, 1995.

[31] Jayaram, S. and A. Myklebust, "Device-Independent Programming Environments for CAD/CAM Software Creation," *Computer-Aided Design*, Vol 25, No. 2, pp. 94-105, February 1993.

[32] Angster, S. Jayaram, S. and D. Hutton, "Case Studies on the Use of Virtual Reality for an Integrated Design and Manufacturing System," *Proceedings of DETC'97, 1997 ASME Design Engineering Technicai Conferences*, Sacramento, CA, September 14-17, 1997.

[33] Jayaram, S. Connacher, H. I. and K. W. Lyons, "Virtual Assembly using Virtual Reality Techniques," *Computer-Aided Design*, Vol. 29, No. 8, pp. 575-584, August 1997.

[34] Gupta, R., and D. Zelzter, "Prototyping and Design for Assembly Analysis using Multi-model Virtual Environments," *Proceedings of the 15th ASME International Computers in Engineering Conference*, Boston, MA, September 17-21, 1995.

[35] Jacobsen, S. C., Smith, F. M., Backman, D. K. and E. K. Iversen, "High Performance, High Dexterity, Force Reflective Teleoperator II," *Proceedings of ANS Topical Meeting on Robotics and Remote Systems*, Albuquerque, NM, February 1991.

[36] Hancock, D. "Prototyping the Hubble Fix," *IEEE Spectrum*, pp. 22-29, October 1993.

[37] Rosenblum, L. J., "Mission Visualization for Planning and Training," *IEEE Computer Graphics and Applications*, Vol. 15, pp. 12-14, September 1995.

[38] Yagawa, G., Kawai, H., Yoshimura, S., and A. Yoshioka, "Mesh-Invisible Finite Element Analysis System in a Virtual Reality Environment," *Computer Modeling and Simulation in Engineering*, Vol. 1, No. 2., pp. 289-314, May 1996.

[39] "FEA Enters the Virtual World," *Design News*, Vol. 51, p. 55, July 22, 1996.

[40] Yeh, T. P. and J. M. Vance, "Combining MSC/NASTRAN, Sensitivity Methods, and Virtual Reality to Facilitate Interactive Design," *Finite Elements in Analysis and Design*, Vol. 26, pp. 161-169, 1997.

[41] Haase, H. and T. Preβ, "Improved Interaction and Visualization of Finite Element Data for Virtual Prototyping," *Proceedings of DETC'97, 1997 ASME Design Engineering Technical Conference*, Sacramento, CA. September 14-17, 1997.

[42] Lamousin, H. J. and W. N. Waggenspack, Jr., "NURBS-Based Free-Form Deformations," *IEEE Computer Graphics and Applications*, Vol. 14, No. 6, pp. 59-65, November 1994.

[43] Rizai, M. N. and J. E. Bernard, "An Efficient Method to Predict the Effect of Design Modifications," *Journal of Mechanisms, Transmission, and Automation in Design*, Vol. 109, pp. 377-384, September 1987.

[44] *WorldToolKit Reference Manual, Release 6*, Sense8 Corporation, Mill Valley, CA, 1996.

[45] *MSC/NASTRAN User's Manual Version 68*, The MacNeal Schwendler Corporation, Los Angles, CA, 1991.

[46] *I-DEAS Master Series Student Guide*, Structural Dynamics Research Corporation, Milford, Ohio, 1994.

[47] Anand, V. B., *Computer Graphics and Geometric Modeling for Engineers*, John Wiley & Sons, New York, 1993.

[48] Piegl, L., "On NURBS: A Survey," *IEEE Computer Graphics & Applications*, pp. 55-71, January 1991.

[49] Versprille, K. J., *Computer-Aided Design Applications of the Rational B-Spline Approximation Form*, doctoral dissertation, Syracuse University, Syracuse, N.Y., 1975.

[50] *The Internal Graphic Exchange Specification (IGES)* Version 5.2, ANSI Y14.26M, U.S. Product Data Association (US PRO), Fairfax, VA, USA, 1993.

[51] *Programmer's Hierarchical Interactive Graphics System (PHIGS)*, ISO/IEC 9592-4, National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, 1992.

[52] *Standard for the Exchange of Product Model Data (STEP)*, ISO 10303, Part 42: Geometric and Topological Representation," ISO 10303-42, ISO Secretariat, National Institute of Standards and Technology (NIST), Gaithersburg, MD, 1994.

[53] Sederberg, T. W. and S. R. Parry, "Free-Form Deformation of Solid Geometric Models," *Computer Graphics*, Vol. 20, No. 4, pp. 151-160, August 1986.

[54] Coquillart, S., "Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modeling," *Computer Graphics*, Vol. 24, No. 4, pp. 181-187, August 1990.

[55] Griessmair, J. and W. Purgathofer, "Deformation of Solids with Trivariate B-Splines," *Proceeding of Eurographics 89*, Elsevier Science Publishers, North-Holland, pp. 137-148, 1989.

[56] Lasser, D., "Rational Tensor Product Bézier Volumes," *Computer Math. Applic.*, Vol. 29, No. 8, pp. 95-108, 1995.

[57] Hsu, W. M., Hughes, J. F. and H. Kaufman, "Direct Manipulation of Free-Form Deformations," *Computer Graphics*, Vol. 26, No. 2, pp. 177-184, July 1992.

[58] Rogers, D. F. and S. G. Scatterfield, "Dynamic B-Spline Surfaces," *Proceedings of International Conference on Computer Applications in the Automation of Shipyard Operation and Ship Design IV (ICCAS' 82)*, pp. 189-196, June 1982.

[59] Rogers, D. F. and L. A. Adlum, "Dynamic Rational B-Spline Surfaces," *Computer-Aided Design*, Vol. 22, No. 9, pp. 609-616, November 1990.

[60] Au, C. K. and M. M. F. Yuen, "Unified Approach to NURBS Curve Shape Modification," *Computer-Aided Design*, Vol. 27, No. 2, pp. 85-93, 1995.

[61] Tits, A. L., Lawrence, C. and J. L. Zhou, "User's Guide for CFSQP Version 2.4: A C Code for Solving (Large Scale) Constrained Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality Constraints," Electrical Engineering Department, *Institute for System Research TR-94-16rl*, University of Maryland, College Park, 1996.

[62] Noble, B. and J. W. Daniel, *Applied Linear Algebra*, Prentice-Hall, 3rd edition, 1988.

[63] Piegl, L. and W. Tiller, *The NURBS Book*, Springer-Verlag, New York, 1995.

[64] Reklaitis, G. V., Ravindran, A. and K. M. Ragsdell, *Engineering Optimization Methods and Applications*, John Wiley & Sons, New York, 1983.

[65] Ding, Y., "COMPENDIUM Shape Optimization of Structures: A Literature Survey," *Computers and Structures*, Vol. 24, No. 6, pp. 985-1004, 1986.

[66] Haftka, R. T. and R. V. Grandhi, "Structural Shape Optimization - A Survey," *Computer Methods in Applied Mechanics and Engineering*, Vol. 57, pp. 91-106, 1986.

[67] Zienkiewicz, O. C. and J. S. Campbell, "Shape Optimization and Sequential Linear Programming," *Optimum Structural Design*, Gallagher, R. H. and O. C. Zienkiewicz (Eds) Wiely, New York, pp. 109-126, 1973.

[68] Wang, S. Y., Sun, T. B., and R. H. Gallagher, "Sensitivity Analysis in Shape Optimization of Continuum Structures," *Computers and Structures*, Vol. 20, No. 5, pp. 855-867, 1985.

[69] Somayajula, G and J. E. Bernard, "Grid Sensitivity Analysis," *Finite Element in Analysis and Design*, Vol. 7, pp. 307-315, 1991.

[70] Cheu, T. C., "Sensitivity Analysis and Shape Optimization of Axisymmetric Structures," *International Journal for Numerical Methods in Engineering*, Vol. 28, pp. 95-108, 1989.

[71] Imam, M. H., "Three-Dimensional Shape Optimization," *International Journal for Numerical Methods in Engineering*, Vol. 18, pp. 661-673, 1982.

[72] Bhavikatti, S. S. and C. V. Ramakrishnan, "Optimum Shape Design of Rotating Disk," *Computers and Structures*, Vol. 11, pp. 397-401, 1980.

[73] Prasad, B and J. F. Emerson, "Optimal Structural Remodeling of Multi-Objective Systems," *Computers and Structures*, Vol. 18, No. 4, pp. 619-628, 1984.

[74] Pedersen, P and C. L. Laursen, "Design for Minimum Stress Concentration by Finite Elements and Linear Programming," *Journal of Structural Mechanics*, Vol. 10, pp. 375-391, 1982-83.

[75] Yang, R. J. and K. K. Choi, "Accuracy of Finite Element Based Design Sensitivity Analysis," *Journal of Structural Mechanics*, Vol. 13, pp. 223-239, 1985.

[76] Luchi, M. L., Poggialini, A and F. Persiani, "An Interactive Optimization Procedure Applied to the Design of Gas Turbine Discs," *Computers and Structures*, Vol. 11, pp. 629-637, 1980.

[77] Weck, M. and P. Steinke, "An Efficient Technique in Shape Optimization," *Journal of Structural Mechanics*, Vol. 11, pp. 433-449, 1983-84.

[78] Braibant, V. and C. Fleury, "Shape Optimal Design Using B-Spline," *Computer Methods in Applied Mechanics and Engineering*, Vol. 44, pp. 247-267, 1984.

[79] Yao, T. M. and K. K. Choi, "3D Shape Optimal Design and Automatic Finite Element Regridding," *International Journal for Numerical Methods in Engineering*, Vol. 28, pp. 369-384, 1989.

[80] Schramm, U. and W. D. Pilkey, "The Coupling of Geometric Descriptions and Finite Elements Using NURBS - A Study in Shape Optimization," *Finite Element in Analysis and Design*, Vol. 15, pp. 11-34, 1993.

[81] Kodiyalam, S., Kumar, V. and P. M. Finnigan, "Constructive Solid Geometry Approach to Three-Dimensional Structural Shape Optimization," *AIAA Journal*, Vol. 30, No. 5, pp. 1408-1415, May 1992.

[82] Botkin, M. E., "Shape Design Modeling Using Fully Automatic Three-Dimensional Mesh Generation," *Finite Element Analysis and Design*, Vol. 10, pp. 165-181, 1991.

[83] Hardee, H, Chang, K. H., K. K. Choi, Yu, X., and I. Grindeanu, "A CAD-Based Design Sensitivity Analysis and Optimization for Structural Shape Design Application," *6th Annual AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Bellevue, WA, pp. 77-87, 1996.

[84] Chen, S. and D. A. Tortorelli, "Three-Dimensional Shape Optimization With Variational Geometry," *6th Annual AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Bellevue, WA, pp. 99-113, 1996.

[85] Hou, J. W. and J. S. Sheen, "On the Design Velocity Field in the Domain and Boundary Methods for Shape Optimization," *AIAA Paper No. 88-2338*, pp. 1032-1040, 1988.

[86] Choi, K. K. and K. H. Chang, "A Study of Design Velocity Field Computation for Shape Optimal Design," *Finite Element in Analysis and Design*, Vol. 15, pp. 317-341, 1994.

[87] Yang, R. J. "A Three-Dimensional Shape Optimization System - SHOP3D," *Computers and Structures*, Vol. 31, No. 6, pp. 881-890, 1989.

[88] Botkin, M. E., "Shape Optimization of Plate and Shell Structures," *AIAA Journal*, Vol. 20, No. 2, pp. 268-273, 1982.

[89] Yang, R. J. and M. E. Botkin, "A Modular Approach for Three-Dimensional Shape Optimization," *AIAA Journal*, Vol. 25, No. 3, pp. 492-497, 1987.

[90] Chang, K. H. and K. K. Choi, "A Geometry Based Parametrization Method for Shape Design of Elastic Solids," *Mech. Struct. and Machines*, Vol. 20, No. 2, pp. 215-252, 1992.

[91] Yao, T. M. and K. K. Choi, "Shape Optimal Design of An Arch Dam," *ASME Journal of Structural Engineering*, Vol. 115, No. 9, pp. 2401-2405, 1989.

[92] Rajan, S. D. and Belegundu, "Shape Optimization Approach Using Fictitious Loads," *AIAA Journal*, Vol. 27, No. 1, pp. 102-107, 1989.

[93] Belegundu, A. D. and S. D. Rajan, "A Shape Optimization Approach Based on Natural Design Variable and Shape Functions," *Computer Methods in Applied Mechanics and Engineering*, Vol. 66. pp. 89-106, 1988.

[94] Haug, E. J., Choi, K. K. and V. Komkov, *Design Sensitivity Analysis of Structural Systems*, Academic Press, New York, 1986.

[95] Yeh, T. P. and J. M. Vance, "Interactive Design of Structural Systems in a Virtual Environment," *Proceedings of 24th Midwest Mechanics Conference*, Iowa State University, Ames, IA, pp. 185-187, 1995.

[96] Adelman, H. M. and R. T. Haftka, "Sensitivity Analysis of Discrete Structural System," *AIAA Journal*, Vol. 24, No. 5, pp. 823-832, 1986.

[97] Kwak, B. M. "A Review on Shape Optimal Design and Sensitivity Analysis," *Structural Engineering/Earthquake Engineering*, Vol. 10, No. 4, pp. 159s-174s, January 1994.

[98] Choi, K. K. and H. G. Seong, "A Domain Method for Shape Design Sensitivity Analysis," *Computer Methods in Applied Mechanics and Engineering*, Vol. 57, pp. 1-15, 1986.

[99] Yang, R. J. and M. E. Botkin, "Accuracy of the Domain Material Derivative Approach to Shape Design Sensitivities," *AIAA Journal*, Vol. 25, No. 12, pp. 1606-1610, 1987.

[100]Chen, J. L. and J. S. Ho, "A Comparative Study of Design Sensitivity Analysis by Using Commercial Finite Element Programs," *Finite Elements in Analysis and Design*, Vol. 15, pp. 189-200, 1994.

[101]Lahey, R. S., "Introduction of Design Sensitivity Into A General Purpose Finite Element Program," *Proceedings of the 1983 ASME International Computers in Engineering Conference*, Vol. 3 pp. 211-216, 1983.

[102]Yang, R. J., Dewhirst, D. L., Allison, J. E. and A. Lee, "Shape Optimization of Connecting Rod Pin End Using a Generic Model," *Finite Elements in Analysis and Design*, Vol. 11, pp. 257-264, 1992.

[103]Vance, J. M. and J. E. Bernard, "Using Padé Approximation and Curve-fitting to Approximate Eigenvalue and Eigenvectors for Large Design Changes," *ASME Journal of Mechanical Design, Technical Note*, Vol. 113, No. 1, pp. 151-153, March 1996.

[104] Vance, J. M. and J. E. Bernard, "Approximating Eigenvectors and Eigenvalues Across a Wide Range of Design," *Finite Elements in Analysis and Design*, Vol. 14, pp. 403-414, 1993.

[105]Chargin, M. K., Raasch, I, and Bruns, R. and D. Deuermeyer, "General Shape Optimization Capability," *Finite Elements in Analysis and Design*, Vol. 7, pp. 343-354, 1991.

[106]Vanderplaats, G. N., *Numerical Optimization Techniques for Engineering Design: With Applications*, McGraw-Hill, New York, 1984.

[107]Starkey, J. M. and J. E. Bernard, "A Constraint Function Technique for Improved Structural Dynamics," *ASME Journal of Vibration, Acoustics, Stress, and Reliability in Design*, Vol. 108, pp. 101-106, January 1986.

[108]Fox, R. L. and M. P. Kapoor, "Rates of Change of Eigenvalues and Eigenvectors," *AIAA Journal*, Vol. 6, No. 12, pp. 2426-2429, 1968.

[109]Ames, A. L., Nadeau, D. R. and J. L. Moreland, *The VRML Sourcebook*, John Wiley & Sons, Inc., New York, 1996.

[110]Moore, G. J., *MSC/NASTRAN Design Sensitivity and Optimization, User's Guide*, Version 67, The MacNeal Schwendler Corporation, Los Angles, CA, 1992.